

CA-IDMS[®]/Schema Mapper

User Guide

15.0



Computer Associates

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

THIS DOCUMENTATION MAY NOT BE COPIED, TRANSFERRED, REPRODUCED, DISCLOSED, OR DUPLICATED, IN WHOLE OR IN PART, WITHOUT THE PRIOR WRITTEN CONSENT OF CA. THIS DOCUMENTATION IS PROPRIETARY INFORMATION OF CA AND PROTECTED BY THE COPYRIGHT LAWS OF THE UNITED STATES AND INTERNATIONAL TREATIES.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

THE USE OF ANY PRODUCT REFERENCED IN THIS DOCUMENTATION AND THIS DOCUMENTATION IS GOVERNED BY THE END USER'S APPLICABLE LICENSE AGREEMENT.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227.7013(c)(1)(ii) or applicable successor provisions.

First Edition, December 2000

© 2000 Computer Associates International, Inc.
One Computer Associates Plaza, Islandia, NY 11749
All rights reserved.

All trademarks, trade names, service marks, or logos referenced herein belong to their respective companies.

Contents

How to Use This Manual	vii
Chapter 1. General Information	1-1
1.1 Create Data Structure Diagrams Automatically	1-4
1.2 Processing Environment	1-5
1.3 Powerful Layout and Format Capabilities	1-6
1.4 CA-IDMS/Schema Mapper's Inputs	1-7
1.4.1 Parameter Statements	1-7
1.4.2 CA-IDMS Dictionary	1-8
1.5 CA-IDMS/Schema Mapper's Four Outputs	1-10
1.5.1 Data Structure Diagram	1-10
1.5.2 Automatic Layout and Format by Default	1-10
1.5.3 Tailoring the Layout and Format	1-10
1.5.4 Layout Parameters--Optional INCLUDE AREA Statement	1-11
1.5.5 Optional LOCATE Statement	1-11
1.5.6 Format Parameters--Optional Statements	1-11
1.5.7 Wallpapering--Technique Used to Assemble a Data Structure Diagram	1-12
1.5.8 How Large Will Your Diagrams Be?	1-12
1.5.9 Transfer File	1-14
1.5.10 Using the Transfer File	1-14
1.5.11 Ways to Use the Transfer File	1-16
1.5.12 Cross-Reference Report	1-19
1.5.13 Tailoring the Cross-Reference Report with Optional Parameters	1-19
1.5.14 Audit Report	1-20
Chapter 2. Parameters	2-1
2.1 Using CA-IDMS/Schema Mapper's Parameters	2-4
2.1.1 PROCESS Statement	2-4
2.1.2 OPTIONS Statement	2-4
2.1.3 CHARDEF Statement	2-4
2.1.4 DRECLINE Statement	2-5
2.1.5 XRECLINE Statement	2-5
2.1.6 DSETLINE Statement	2-5
2.1.7 XSETLINE Statement	2-5
2.1.8 INCLUDE AREA Statement	2-5
2.1.9 LOCATE Statement	2-5
2.2 Notations Conventions and Syntax Rules	2-6
2.3 PROCESS Statement	2-10
2.3.1 PROCESS Syntax	2-10
2.4 OPTIONS Statement	2-12
2.4.1 OPTIONS Syntax	2-12
2.4.2 Specifying the Positions of Index	2-12
2.4.3 Changing Page Dimensions for Paper Size and Printer Flexibility	2-13
2.4.4 Defining Border Space Between Record Blocks	2-14
2.4.5 Compressing Unused Space in the Data Structure Diagram	2-14
2.4.6 Set Linkage with Areas not Represented in Diagram	2-14
2.5 CHARDEF Statement	2-15

2.5.1	CHARDEF Syntax	2-15
2.5.2	Specifying Print Characters Used to Draw Record Blocks	2-16
2.5.3	Specifying Print Characters Used to Draw Set Connections	2-16
2.5.4	Specifying Print Characters Used to Draw Arrows	2-17
2.5.5	Specifying Print Characters Used to Draw Index Lines	2-18
2.6	DRECLINE Statement	2-19
2.6.1	DRECLINE Syntax	2-20
2.6.2	How to Use the DRECLINE Statement	2-21
2.6.3	Graphic Considerations	2-21
2.6.4	Creating Blank Fields and Blank Lines In Record Blocks	2-21
2.6.5	Placement of Multiple CALC Keys	2-22
2.6.6	Sample DRECLINE Statements and Resulting Record Blocks	2-23
2.7	XRECLINE Statement	2-27
2.7.1	XRECLINE Syntax	2-27
2.7.2	How to Use the XRECLINE Statement	2-29
2.7.3	Creating Blank Fields and Blank Lines In Record Descriptions	2-29
2.7.4	Graphic Considerations	2-29
2.7.5	Placement of Multiple CALC Keys	2-29
2.7.6	Sample XRECLINE Statements and Resulting Record Descriptions	2-30
2.8	DSETLINE Statement	2-33
2.8.1	DSETLINE Syntax	2-33
2.8.2	How to Use the DSETLINE Statement	2-34
2.8.3	Creating Blank Fields and Blank Lines in Set Descriptions	2-34
2.8.4	Graphic Considerations	2-34
2.8.5	Placement of Multiple Sort or Index Keys	2-35
2.8.6	Sample DSETLINE Statements and Resulting Set Descriptions	2-36
2.9	XSETLINE Statement	2-38
2.9.1	XSETLINE Syntax	2-38
2.9.2	How to Use the XSETLINE Statement	2-40
2.9.3	Creating Blank Fields and Blank Lines in Set Descriptions	2-40
2.9.4	Graphic Considerations	2-40
2.9.5	Placement of Multiple Sort or Index Keys	2-40
2.9.6	Sample XSETLINE Statements and Resulting Set Descriptions	2-42
2.10	INCLUDE AREA Statement	2-46
2.10.1	INCLUDE AREA Syntax	2-46
2.10.2	When to Use the INCLUDE AREA Statement	2-46
2.11	LOCATE Statement	2-47
2.11.1	LOCATE Syntax	2-47
2.11.2	When to Use the LOCATE Statement	2-48
2.11.3	Using the LOCATE Statement Most Efficiently	2-48
2.11.4	Graphic Considerations	2-49
Chapter 3.	System Output	3-1
3.1	Data Structure Diagram	3-4
3.1.1	Basic Components of the Data Structure Diagram	3-4
3.1.2	Wallpapering: Technique Used to Assemble a Data Structure Diagram	3-6
3.1.3	How Large Will Your Diagrams Be?	3-9
3.2	Transfer File	3-10
3.2.1	Transfer File Statements	3-10
3.3	Cross-Reference Report	3-12
3.3.1	Cross-Reference Report Field Descriptions	3-12

3.4 Audit Report	3-15
Chapter 4. Examples	4-1
4.1 Example 1	4-4
4.2 Example 2	4-7
4.3 Example 3	4-9
4.4 Example 4	4-13
4.5 Example 5	4-15
Chapter 5. Operations	5-1
5.1 System Requirements	5-4
5.1.1 Storage Requirements	5-4
5.2 OS/390 Environments	5-5
5.2.1 OS/390 JCL	5-5
5.2.2 Key to OS/390 JCL	5-5
5.3 VSE/ESA Environments	5-8
5.3.1 VSE/ESA File Assignments	5-8
5.3.2 VSE/ESA JCL	5-8
5.3.3 Key to VSE/ESA JCL	5-10
5.4 VM/ESA Environments	5-12
5.4.1 VM/ESA EXEC	5-12
5.4.2 Key to VM/ESA EXEC	5-13
Chapter 6. Messages	6-1
Glossary	X-1
Index	X-3

How to Use This Manual

Purpose

This guide provides the information needed to run CA-IDMS/Schema Mapper. In addition, the many features that CA-IDMS/Schema Mapper offers are documented to assist you.

Organization

Chapter	Description
1	Presents a summary of CA-IDMS/Schema Mapper capabilities and gives an overview of the inputs and outputs.
2	Describes how to use and gives detailed information about CA-IDMS/Schema Mapper's parameter statements.
3	Gives an overview and describes the components of each output created by CA-IDMS/Schema Mapper.
4	Gives examples of output created with CA-IDMS/Schema Mapper. It gives examples of how optional parameters affect the diagrams and reports CA-IDMS/Schema Mapper creates.
5	Details CA-IDMS/Schema Mapper operations, including system requirements and storage requirements.
6	Provides a list of all informative, error, and warning messages generated by CA-IDMS/Schema Mapper, along with reasons for occurrence and suggested actions to be taken.
Glossary	Gives a list of key terms with their meanings used in this user guide.
Index	Provides an alphabetical list of CA-IDMS/Schema Mapper concepts with their locations in the user guide.

Name	Contents
CA-IDMS Installation and Maintenance Guide	An installation guide is provided to use as a reference tool and gives complete information about the installation of the products.

Chapter 1. General Information

1.1 Create Data Structure Diagrams Automatically	1-4
1.2 Processing Environment	1-5
1.3 Powerful Layout and Format Capabilities	1-6
1.4 CA-IDMS/Schema Mapper's Inputs	1-7
1.4.1 Parameter Statements	1-7
1.4.2 CA-IDMS Dictionary	1-8
1.5 CA-IDMS/Schema Mapper's Four Outputs	1-10
1.5.1 Data Structure Diagram	1-10
1.5.2 Automatic Layout and Format by Default	1-10
1.5.3 Tailoring the Layout and Format	1-10
1.5.4 Layout Parameters--Optional INCLUDE AREA Statement	1-11
1.5.5 Optional LOCATE Statement	1-11
1.5.6 Format Parameters--Optional Statements	1-11
1.5.7 Wallpapering--Technique Used to Assemble a Data Structure Diagram	1-12
1.5.8 How Large Will Your Diagrams Be?	1-12
1.5.9 Transfer File	1-14
1.5.10 Using the Transfer File	1-14
1.5.11 Ways to Use the Transfer File	1-16
1.5.12 Cross-Reference Report	1-19
1.5.13 Tailoring the Cross-Reference Report with Optional Parameters	1-19
1.5.14 Audit Report	1-20

With CA-IDMS/Schema Mapper, you can automatically create data structure diagrams (graphic representations) of CA-IDMS schemas and subschemas from data in the CA-IDMS dictionary. You can let CA-IDMS/Schema Mapper determine everything in the data structure diagram, from record block and set description formats to diagram layout, or you can define some or all of the diagram's format and layout using parameter statements. CA-IDMS/Schema Mapper gives you automatic diagram creation and flexibility.

1.1 Create Data Structure Diagrams Automatically

Data structure diagrams are invaluable to CA-IDMS users who need clearly defined graphic representations of schema and subschema definitions. In fact, accurate pictures of a database's structure are such powerful tools that programmers and DBAs refer to them every day.

Creating and maintaining data structure diagrams requires a lot of time. Every change to a database's structure, however small, means tedious and detailed redrawing of many or all of the records and set connections. CA-IDMS/Schema Mapper is an easy-to-use software tool that eliminates the tedious drawing and redrawing usually associated with creating and maintaining data structure diagrams.

CA-IDMS/Schema Mapper automates the once time-consuming process of drawing data structure diagrams.

1.2 Processing Environment

CA-IDMS/Schema Mapper runs in OS/390, VSE/ESA, and VM/ESA environments with the following product :

- CA-IDMS 15.0.

1.3 Powerful Layout and Format Capabilities

Creating data structure diagrams with CA-IDMS/Schema Mapper is a parameter-driven process that lets you use these powerful capabilities:

- **Automatic diagram layout and default format**--A well-planned automatic diagram layout and default format allow you to create data structure diagrams with a minimal amount of time and effort. To generate the automatic default diagram, you supply a single parameter statement.
- **User-defined format**--You can tailor the format of record blocks, sets, and indexes by using several parameter options. You can specify which fields are included, the order of the fields, and the line lengths.
- **User-defined diagram layout**--You can tailor the layout of your diagram with optional parameters. You can select areas to be represented by the diagram, specify the placement of a few or all of the record blocks in the diagram, and you can specify the amount of space between record blocks.
- **Your printer as a graphics tool**--You do not need a special printer to create data structure diagrams with CA-IDMS/Schema Mapper. The characters used to print the default diagram are supported by most printers. And if your printer does not have some of the default characters, CA-IDMS/Schema Mapper has a parameter statement that allows you to specify any of the characters used to draw the diagram. CA-IDMS/Schema Mapper provides for any character set that your printer might use.
- **Cross-Reference Report**--This report provides you with a useful tool for quickly finding the locations of all records, sets, and indexes in a large data structure diagram, and it also provides information on the definitions of records and sets.
- **Audit Report**--This report helps you monitor program execution and identify problems.

1.4 CA-IDMS/Schema Mapper's Inputs

CA-IDMS/Schema Mapper uses parameter statements and the CA-IDMS dictionary as input. The CA-IDMS/DB ANALYZER Statistic File can also be used (see Exhibit 1.1).

1.4.1 Parameter Statements

CA-IDMS/Schema Mapper has several parameter statements for creating and tailoring data structure diagrams. The PROCESS statement is required for CA-IDMS/Schema Mapper execution, the other statements are optional. You can combine the statements in a variety of ways to produce diagrams that meet your needs. The chart in Exhibit 1.2 summarizes each of the parameter statements.

1.4.2 CA-IDMS Dictionary

CA-IDMS/Schema Mapper reads the definition of the schema or subschema you want to represent from the CA-IDMS dictionary.

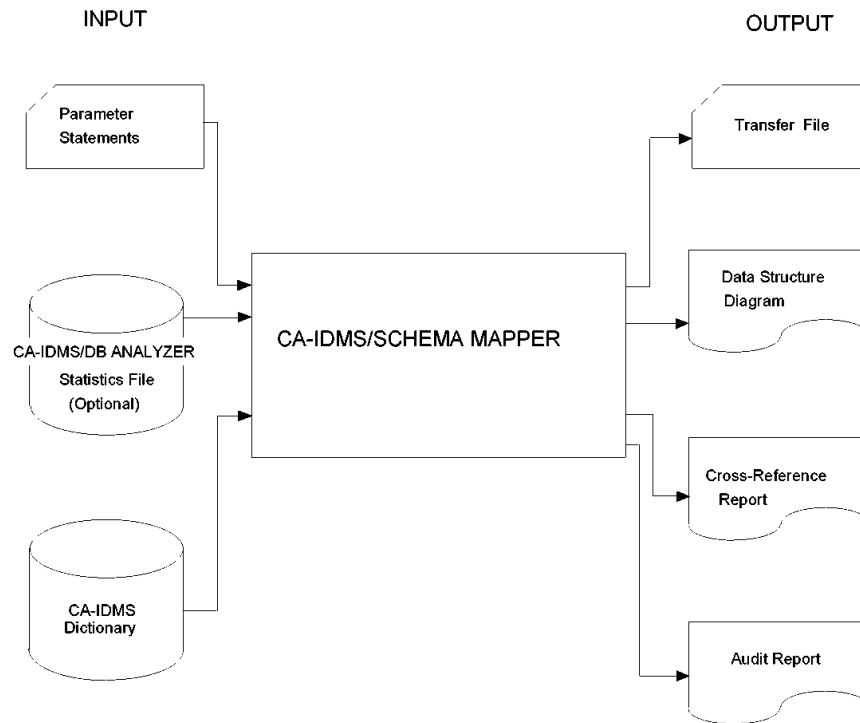


Exhibit 1.1: CA-IDMS/Schema Mapper Process

Statement	Required?	Description
PROCESS	Yes	Identifies the schema or subschema to be represented by a data structure diagram.
OPTIONS	No	With OPTIONS, you can specify page dimensions, the amount of border space surrounding record blocks, whether or not unused space is compressed from the diagram, and how indexes are drawn in the diagram.
CHARDEF	No	Specifies options for fine-tuning the diagram. You can define which characters are used to draw record blocks, set connections, set turns, and arrows.
DRECLINE	No	Specifies the textual format of record block descriptions in the data structure diagram.
XRECLINE	No	Specifies the textual format of record blocks in the Cross-Reference Report.
DSETLINE	No	Specifies the textual format of both sets and indexes in the data structure diagram.
XSETLINE	No	Specifies the textual format of both sets and indexes in the Cross-Reference Report.
INCLUDE	No	Identifies a specific area from the schema or subschema to be represented by a data structure diagram.
LOCATE	No	Positions record blocks in the diagram.

Exhibit 1.2: Parameter Statement Description

1.5 CA-IDMS/Schema Mapper's Four Outputs

CA-IDMS/Schema Mapper uses the input from parameter statements, the CA-IDMS dictionary, and the CA-IDMS/DB ANALYZER Statistics File to give you four outputs: the data structure diagram, the Transfer File, the Cross-Reference Report, and the Audit Report.

1.5.1 Data Structure Diagram

The first output, the data structure diagram, is a graphic representation of the specified schema or subschema definition. CA-IDMS/Schema Mapper's easy-to-use, flexible parameters let you tailor both the layout and the format of the data structure diagram to meet your individual needs. The layout of the diagram and the formats of the record blocks, sets, and indexes can be defined in three ways:

- Automatic layout and default format values
- Total user control of both the layout and format
- A combination of the two.

1.5.2 Automatic Layout and Format by Default

The data structure diagram that is generated with automatic layout and default formats is quite likely to meet all of your requirements, and it is very easy to generate.

To generate a diagram automatically and by default, you specify the name of the schema or subschema to be represented using the PROCESS statement, the rest is up to CA-IDMS/Schema Mapper. The normal information found in the CA-IDMS dictionary and CA-IDMS/Schema Mapper's sophisticated record layout algorithm are used to draw the diagram.

CA-IDMS/Schema Mapper automatically places record blocks and indexes in the data structure diagram and draws set connections. Default format specifications identify the text to be included in record blocks, sets, and indexes, with the format for record blocks applying to the records as they appear in the data structure diagram. Default specifications are used to identify sets and indexes by number in the data structure diagram, so the specifications for their formats apply to their appearance in the Cross-Reference Report. By using optional parameter statements, you can include the set text in the data structure diagram and record text in the Cross-Reference Report.

1.5.3 Tailoring the Layout and Format

You may find that the automatic layout and default formats meet all of your needs. Yet, if you want to make changes, you can easily tailor the diagram by using optional parameter statements. Whether you need to make changes because your printer does not support certain characters specified by default, or because you do not like the overall appearance of the diagram--whatever the reason--CA-IDMS/Schema Mapper gives you many options for tailoring the diagram.

Tailoring is likely to be an iterative process. You will probably be tailoring to refine or update a previously-created diagram, and you will invoke CA-IDMS/Schema Mapper several times to produce a finished data structure diagram. Whenever you make any kind of modification to the data structure diagram, you can plan it using the latest diagram printout as a “working copy”.

With careful use of CA-IDMS/Schema Mapper's parameter statements, you can quickly and easily create, modify, and maintain data structure diagrams that accurately reflect the current structure of your database's schemas and subschemas.

1.5.4 Layout Parameters--Optional **INCLUDE AREA** Statement

The **INCLUDE AREA** statement lets you tailor the diagram to represent only those areas that you specify from the schema or subschema. This is helpful when you need a diagram for a particular application or when your original diagram is too large.

1.5.5 Optional **LOCATE** Statement

You can control almost any aspect of a diagram's layout with the **LOCATE** statement. It allows you to place record blocks in locations relative to other record blocks in the diagram. See Chapter 2, “Parameters” on page 2-1 for more information.

1.5.6 Format Parameters--Optional Statements

CA-IDMS/Schema Mapper has four parameter statements for tailoring the format of the diagram.

OPTIONS--Used to specify the page dimensions (in number of lines down and number of characters across) of the diagram, the amount of border space surrounding record blocks, and whether or not unused space is compressed from the diagram.

CHARDEF--Used to specify the graphic components of the diagram that are primarily printer-dependent. These components include characters used for drawing arrows, set connections, and record blocks.

DRECLINE--Used to select the information to be included in record blocks and to specify the sequence of the information within the record blocks, as they appear in the data structure diagram.

DSETLINE--Used to select and format the information to be included in the descriptions of both sets and indexes appearing in the data structure diagram.

1.5.7 Wallpapering--Technique Used to Assemble a Data Structure Diagram

Data structure diagrams are often large, so the printed output spans multiple pages. CA-IDMS/Schema Mapper automatically divides the overall diagram into pieces that are the size of your printer's paper, with each piece being the size of a page. You assemble the pages in columns down the length of the diagram and rows across the width of the diagram (see Exhibit 1.3).

The diagram is printed sequentially, one column of pages after another, beginning with the column that runs down the left side of the diagram (column A). CA-IDMS/Schema Mapper automatically generates column headers at the top of each column to help you find the beginning of the printout for each column.

A unique two-character (alphabetic) page identifier is also automatically printed in the upper right corner of each page: the first character identifies the page's column in the overall diagram, and the second character identifies its row in the overall diagram. The first page of the diagram (the top left corner) is in the first column and the first row and is identified by the characters AA.

To assemble the data structure diagram, manually burst the printout at the column boundaries (at the top of the column headers), align each column to form the horizontal rows, and tape the columns together. You can easily assemble the entire diagram in a few minutes. This process is called wallpapering because the diagrams are usually hung on a wall for viewing.

1.5.8 How Large Will Your Diagrams Be?

The size of the data structure diagram primarily depends on the size of the schema or subschema that is used for the diagram. Other factors that can affect the size of a diagram are:

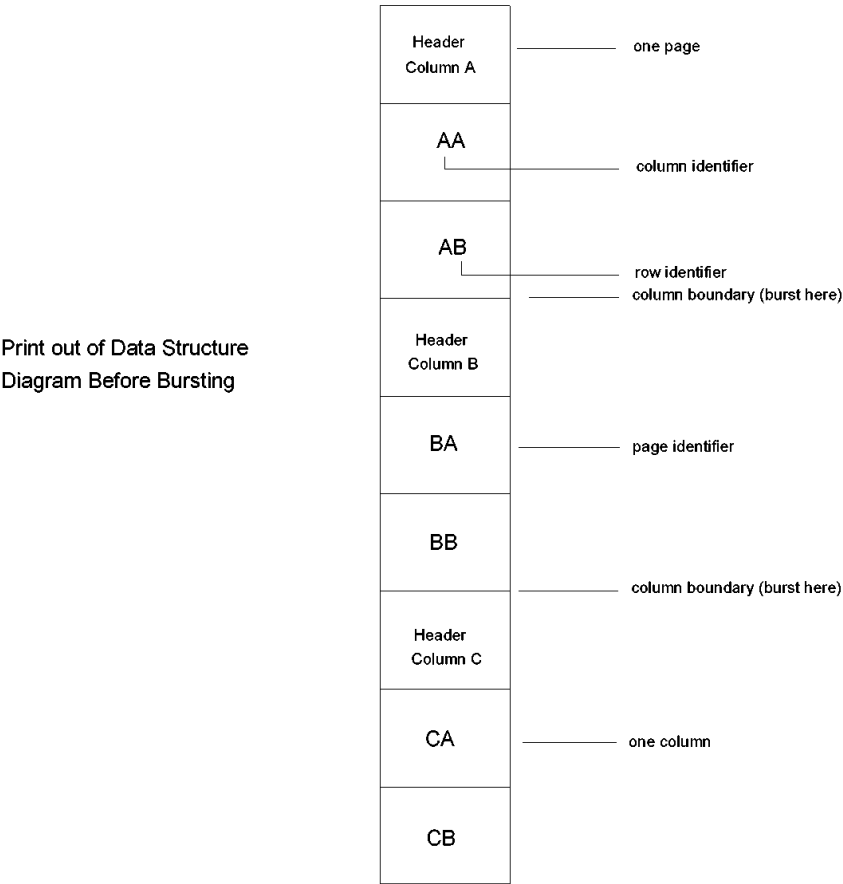
- The configuration of the schema or subschema
- The size of each record block, as determined by the number, length, and organization of fields
- The border space around record blocks.

The size of CA-IDMS/Schema Mapper's CA-IDMS IDMSNWKA diagram may give you an idea of a diagram's size. IDMSNWKA (the subschema for the CA-IDMS Network A database) is eight pages by eight pages (nine feet by seven feet, using 11" by 14" perforated paper). It has 151 records and 227 sets.

Also, a schema containing 45 records, 51 sets, and eight indexes constituted a CA-IDMS/Schema Mapper diagram that was four pages by three pages (four feet by three feet, depending on paper size). This diagram was produced using the default formats and allowing CA-IDMS/Schema Mapper to automatically layout the records.

The maximum size that an assembled CA-IDMS/Schema Mapper data structure diagram can be is 26 pages wide by 26 pages long. You probably won't have any diagrams even close to this size.

If your diagrams of schemas seem too large, you can run CA-IDMS/Schema Mapper for areas or subschemas rather than schemas. Another option, which has good results in reducing the size of diagrams, is to photo-reduce the diagrams.



The Data Structure Diagram
After Bursting and Wallpapering

Header Column A	Header Column B	Header Column C
AA	BA	CA
AB	BB	CB

Exhibit 1.3: Wallpapering the CA-IDMS/Schema Mapper Data Structure Diagram

1.5.9 Transfer File

Every time you invoke CA-IDMS/Schema Mapper, it produces a Transfer File that reflects the layout and format of the corresponding data structure diagram. The Transfer File contains parameter statements copied from the previous input.

The Transfer File always contains a PROCESS statement that specifies the schema or subschema to be represented by a diagram. It also contains LOCATE statements for all of the record blocks in the data structure diagram. CA-IDMS/Schema Mapper writes LOCATEs to the Transfer File for all of the record blocks that you do not manually LOCATE. The Transfer File can also contain OPTIONS, CHARDEF, DRECLINE, XRECLINE, DSETLINE, XSETLINE, and INCLUDE statements, if they were specified in the previous input.

The PROCESS, OPTIONS, CHARDEF, DRECLINE, XRECLINE, DSETLINE, XSETLINE, and INCLUDE statements are exactly the same as the corresponding statements in the preceding input. They are transferred (or copied) to the Transfer File without being modified by CA-IDMS/Schema Mapper.

1.5.10 Using the Transfer File

Creating data structure diagrams with CA-IDMS/Schema Mapper can be either a one-step or a two-step procedure:

Step 1--Execute CA-IDMS/Schema Mapper, specifying the schema or subschema to be represented using the PROCESS statement. You can also use any of the optional statements.

As shown in Exhibit 1.4, Step 1 provides you with CA-IDMS/Schema Mapper's four outputs: the data structure diagram, the Transfer File, the Cross-Reference Report, and the Audit Report. The Transfer File is created as a time-saving device for Step 2.

Step 2--Execute CA-IDMS/Schema Mapper again, making modifications to the input, if needed, by using the optional parameters. You can use the Transfer File as input whenever you execute the second step.

The second step can be iterative: you can repeat it as many times as necessary, depending on whether the preceding data structure diagram meets your needs. Exhibit 1.4 illustrates this process.

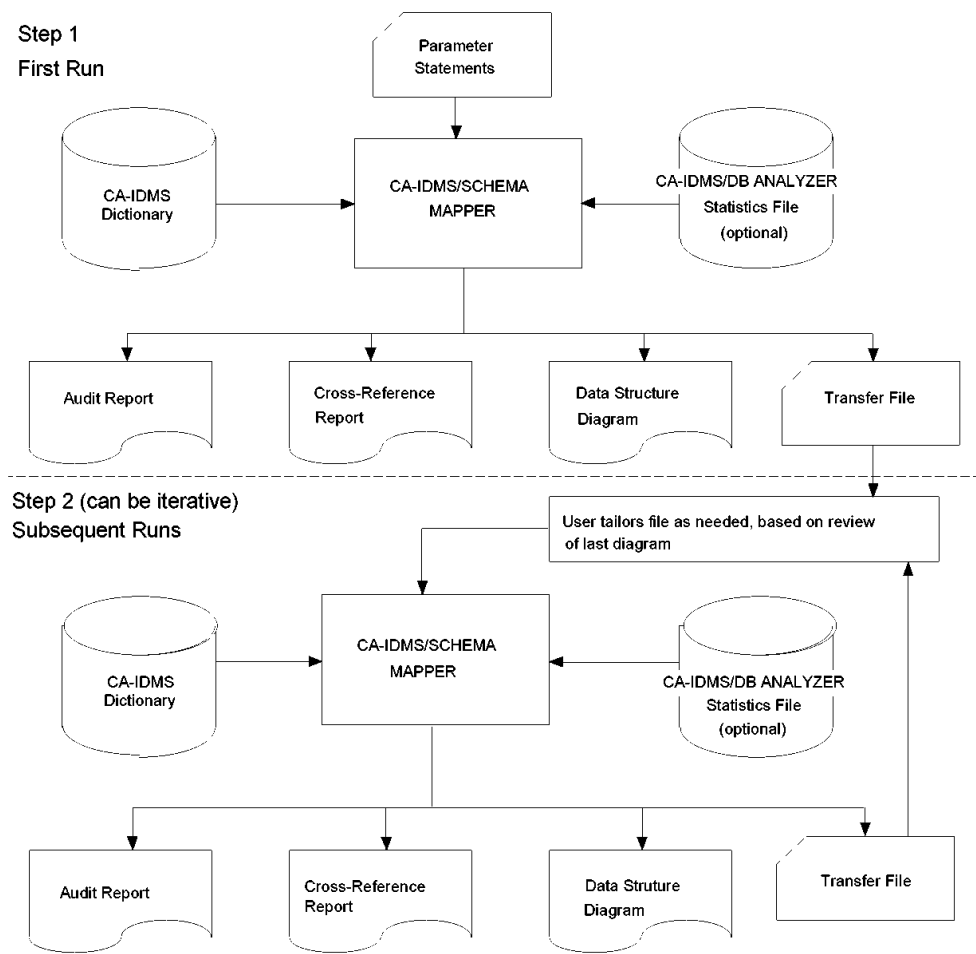


Exhibit 1.4: Creating and Tailoring the CA-IDMS Schema Mapper Data Structure Diagram

1.5.11 Ways to Use the Transfer File

You can use the Transfer File in several ways, as illustrated in Exhibits 1.5a thru 1.5f. You can use the Transfer File to:

- Generate a data structure diagram identical to the preceding one
- Include record blocks, which have been added to the schema or the subschema, in the diagram
- Include only specific areas from a schema or subschema in the diagram
- Modify the layout of the diagram by manually positioning a few or all of the record blocks in the diagram
- Modify the format of the diagram by using any of the independent options that are available
- Modify the layout and the format of the diagram.

Exhibit 1.5a shows the first step for using the Transfer File; Exhibits 1.5b thru 1.5f show the second step for each different way you can use the Transfer File. The first step is always the same, and the second step can be iterative. The exhibits illustrate how easy it is to create and modify data structure diagrams when you use the Transfer File. All you do is specify the parameter statements (marked with arrows in Exhibit 1.5c thru 1.5f). CA-IDMS/Schema Mapper does the rest.

Note: Exhibits 1.5a thru 1.5f show how to use the data structure diagram and Transfer File produced with each execution of CA-IDMS/Schema Mapper. The Cross-Reference Report and Audit Report are also produced with each execution.

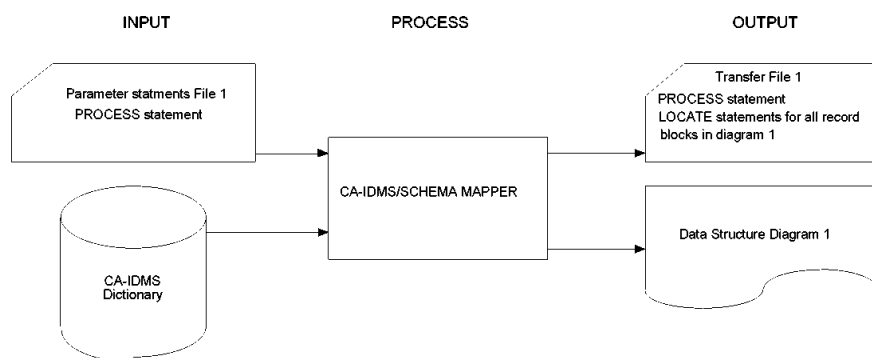


Exhibit 1.5a: Step 1 for Using the Transfer File

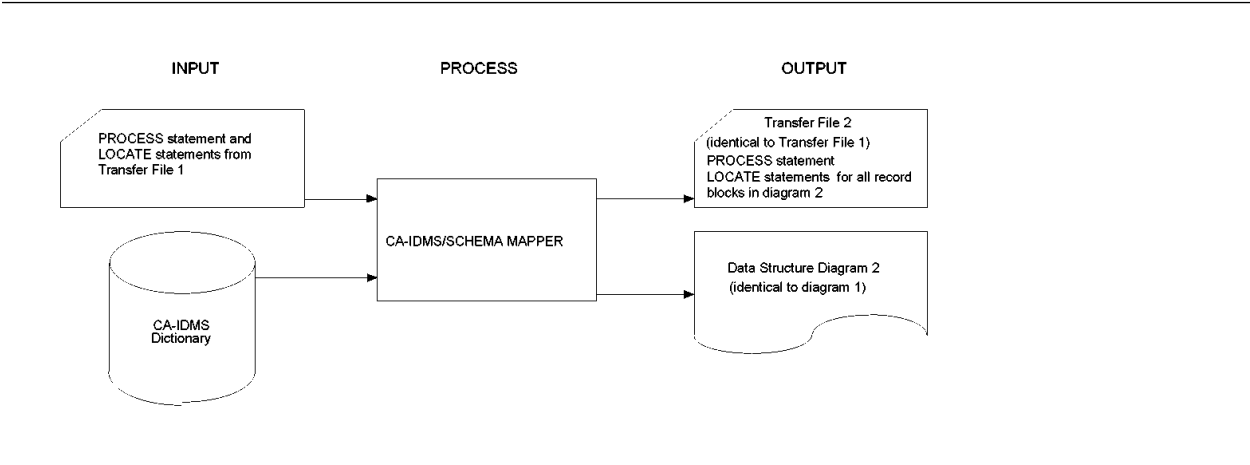


Exhibit 1.5b: Using the Transfer File to Produce the Identical Diagram

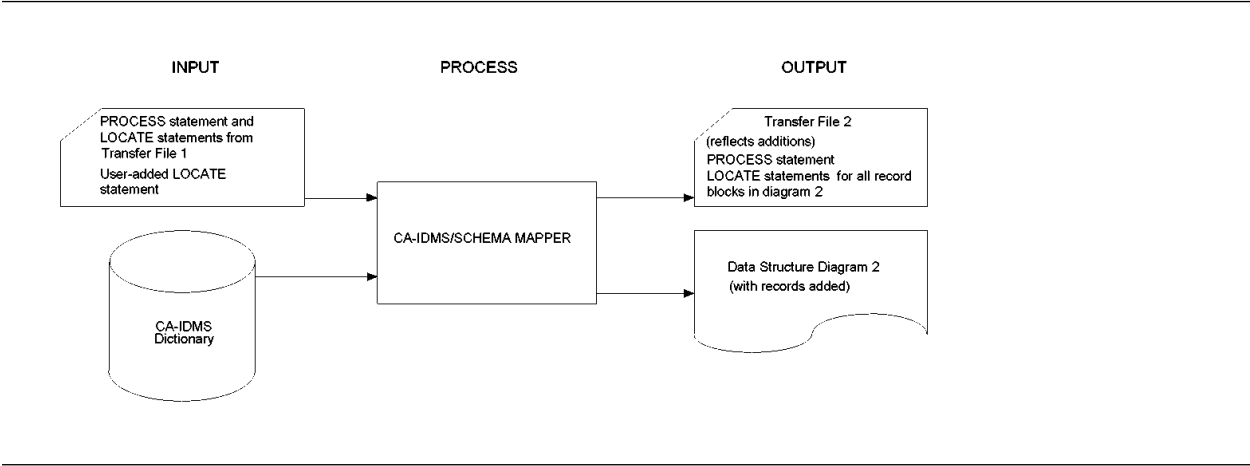


Exhibit 1.5c: Using the Transfer File to Add Record Blocks to the Diagram

1.5 CA-IDMS/Schema Mapper's Four Outputs

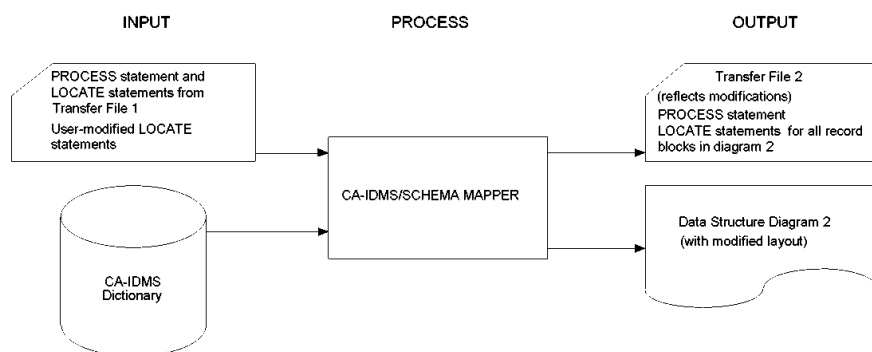


Exhibit 1.5d: Using the Transfer File to Modify the Layout of a Diagram

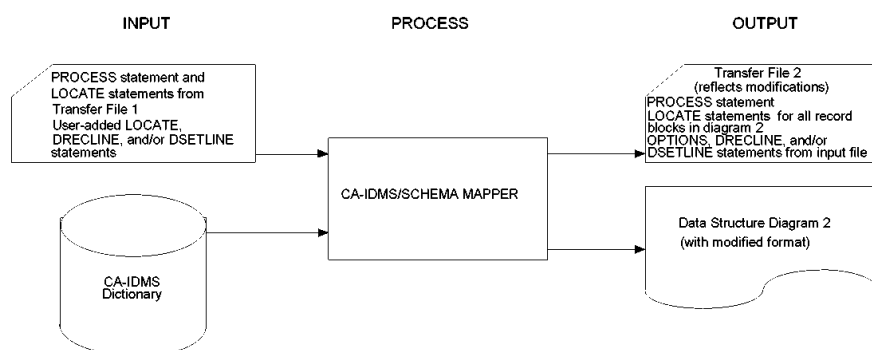


Exhibit 1.5e: Using the Transfer File to Modify the Format of a Diagram

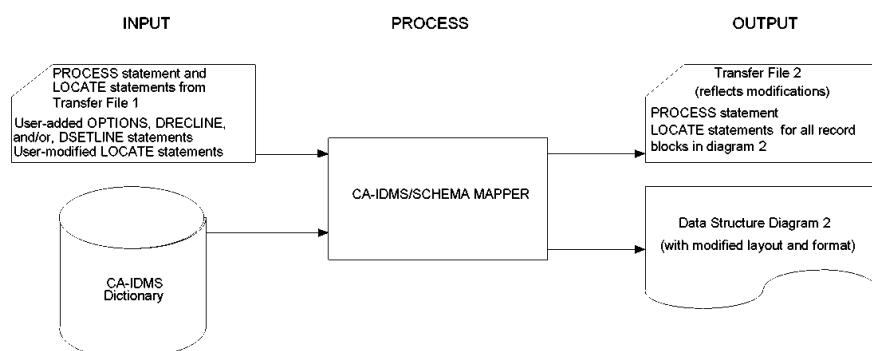


Exhibit 1.5f: Using the Transfer File to Modify the Layout and Format of a Diagram

1.5.12 Cross-Reference Report

The third output is the Cross-Reference Report. It contains the descriptions of records, sets, and indexes and is useful for quickly finding individual record blocks, sets, and indexes in a CA-IDMS/Schema Mapper data structure diagram.

The Cross-Reference Report includes:

- The name and location of each record. The location identifies where the record block can be found on the diagram.
- The name, unique number, description (set and index fields), and location in the data structure diagram of each set and index.
- The names and locations in the diagram of the OWNER and MEMBER records of each set.

1.5.13 Tailoring the Cross-Reference Report with Optional Parameters

The Cross-Reference Report can be tailored to add record descriptions and change the format of both the record and set descriptions. There are two optional parameters for tailoring your reports:

XRECLINE--Used to select the information to be included in record block descriptions and to specify the sequence of the information within the descriptions, as they appear in the Cross-Reference Report.

XSETLINE--Used to select and format the information to be included in the descriptions of both sets and indexes appearing in the Cross-Reference Report.

1.5.14 Audit Report

The fourth output, the Audit Report, contains:

- A summary of all processing performed
- All informative, warning, and error messages generated during program execution
- Suggested remedial actions when appropriate.

Informative messages report processing starts and stops and also provide a list of the parameters used. Warning messages report conflicting parameter data and indicate that CA-IDMS/Schema Mapper performed corrective actions to continue processing. Error messages report the conditions causing termination of diagram processing.

See Chapter 6, “Messages” on page 6-1 for the complete list and explanation of the messages generated by CA-IDMS/Schema Mapper.

When CA-IDMS/Schema Mapper encounters an error condition (generally syntactical), it immediately terminates diagram processing. It continues, however, scanning the parameter statements to detect any additional syntax errors. Because the Audit Report is your only output when CA-IDMS/Schema Mapper encounters an error, the Audit Report is an essential tool for identifying errors that impede diagram creation.

Chapter 2. Parameters

2.1 Using CA-IDMS/Schema Mapper's Parameters	2-4
2.1.1 PROCESS Statement	2-4
2.1.2 OPTIONS Statement	2-4
2.1.3 CHARDEF Statement	2-4
2.1.4 DRECLINE Statement	2-5
2.1.5 XRECLINE Statement	2-5
2.1.6 DSETLINE Statement	2-5
2.1.7 XSETLINE Statement	2-5
2.1.8 INCLUDE AREA Statement	2-5
2.1.9 LOCATE Statement	2-5
2.2 Notations Conventions and Syntax Rules	2-6
2.3 PROCESS Statement	2-10
2.3.1 PROCESS Syntax	2-10
2.4 OPTIONS Statement	2-12
2.4.1 OPTIONS Syntax	2-12
2.4.2 Specifying the Positions of Index	2-12
2.4.3 Changing Page Dimensions for Paper Size and Printer Flexibility	2-13
2.4.4 Defining Border Space Between Record Blocks	2-14
2.4.5 Compressing Unused Space in the Data Structure Diagram	2-14
2.4.6 Set Linkage with Areas not Represented in Diagram	2-14
2.5 CHARDEF Statement	2-15
2.5.1 CHARDEF Syntax	2-15
2.5.2 Specifying Print Characters Used to Draw Record Blocks	2-16
2.5.3 Specifying Print Characters Used to Draw Set Connections	2-16
2.5.4 Specifying Print Characters Used to Draw Arrows	2-17
2.5.5 Specifying Print Characters Used to Draw Index Lines	2-18
2.6 DRECLINE Statement	2-19
2.6.1 DRECLINE Syntax	2-20
2.6.2 How to Use the DRECLINE Statement	2-21
2.6.3 Graphic Considerations	2-21
2.6.4 Creating Blank Fields and Blank Lines In Record Blocks	2-21
2.6.5 Placement of Multiple CALC Keys	2-22
2.6.6 Sample DRECLINE Statements and Resulting Record Blocks	2-23
2.7 XRECLINE Statement	2-27
2.7.1 XRECLINE Syntax	2-27
2.7.2 How to Use the XRECLINE Statement	2-29
2.7.3 Creating Blank Fields and Blank Lines In Record Descriptions	2-29
2.7.4 Graphic Considerations	2-29
2.7.5 Placement of Multiple CALC Keys	2-29
2.7.6 Sample XRECLINE Statements and Resulting Record Descriptions	2-30
2.8 DSETLINE Statement	2-33
2.8.1 DSETLINE Syntax	2-33
2.8.2 How to Use the DSETLINE Statement	2-34
2.8.3 Creating Blank Fields and Blank Lines in Set Descriptions	2-34
2.8.4 Graphic Considerations	2-34
2.8.5 Placement of Multiple Sort or Index Keys	2-35
2.8.6 Sample DSETLINE Statements and Resulting Set Descriptions	2-36

2.9 XSETLINE Statement	2-38
2.9.1 XSETLINE Syntax	2-38
2.9.2 How to Use the XSETLINE Statement	2-40
2.9.3 Creating Blank Fields and Blank Lines in Set Descriptions	2-40
2.9.4 Graphic Considerations	2-40
2.9.5 Placement of Multiple Sort or Index Keys	2-40
2.9.6 Sample XSETLINE Statements and Resulting Set Descriptions	2-42
2.10 INCLUDE AREA Statement	2-46
2.10.1 INCLUDE AREA Syntax	2-46
2.10.2 When to Use the INCLUDE AREA Statement	2-46
2.11 LOCATE Statement	2-47
2.11.1 LOCATE Syntax	2-47
2.11.2 When to Use the LOCATE Statement	2-48
2.11.3 Using the LOCATE Statement Most Efficiently	2-48
2.11.4 Graphic Considerations	2-49

This chapter describes how to use CA-IDMS/Schema Mapper's parameter statements. Except for the **PROCESS** statement, the parameter statements are optional. The optional parameters allow you to tailor the format and layout of the data structure diagram to meet your needs.

2.1 Using CA-IDMS/Schema Mapper's Parameters

The PROCESS statement is the only mandatory parameter statement. The other statements (OPTIONS, CHARDEF, DRECLINE, XRECLINE, DSETLINE, XSETLINE, INCLUDE AREA, and LOCATE) are optional. The optional statements are used when you want to modify the layout or the format of the data structure diagram produced by using a single PROCESS statement.

Creating data structure diagrams with CA-IDMS/Schema Mapper can be either a one-step or a two-step procedure. It is quite likely that you will be satisfied with the diagram created automatically, and you will follow only the first step.

Step 1--Use one PROCESS statement (mandatory) to create a diagram with automatic layout and default format.

Step 2--Assess the diagram and then make desired changes by using one or more of the optional parameter statements.

When you execute CA-IDMS/Schema Mapper, one of its outputs is a Transfer File that contains parameter statements. You can use this file as input to recreate a diagram. You can also modify the parameter statements in the Transfer File to tailor or revise a diagram. Exhibit 2.3 lists all of CA-IDMS/Schema Mapper's parameters.

2.1.1 PROCESS Statement

PROCESS is the only parameter statement needed to execute CA-IDMS/Schema Mapper. Enter PROCESS, specify the schema or subschema to be represented by the diagram, and CA-IDMS/Schema Mapper automatically creates a data structure diagram.

2.1.2 OPTIONS Statement

With the OPTIONS statement, you can include Index Set records in the diagram and change the way indexes are drawn. OPTIONS also controls the dimensions of a page (in number of lines down and characters across) in the data structure diagram, the amount of border space surrounding record blocks, and whether or not unused space is compressed from the diagram.

2.1.3 CHARDEF Statement

With the CHARDEF statement, you can define which characters are to be used to draw record blocks, set connections, set turns, and arrows in the data structure diagram.

2.1.4 DRECLINE Statement

With the DRECLINE statement, you can specify which record fields to include in the data structure diagram record block. A predefined list of record-related fields is included in this chapter of the user guide.

2.1.5 XRECLINE Statement

With the XRECLINE statement, you can control which record fields to include in the Cross-Reference Report. A predefined list of record-related fields is included in this chapter of the user guide.

2.1.6 DSETLINE Statement

With the DSETLINE statement, you control the global format of the text that describes the set in the data structure diagram. A predefined list of set-related fields is included in this chapter of the user guide.

2.1.7 XSETLINE Statement

With XSETLINE, you can determine the global format of the text that describes the set in the Cross-Reference Report. A predefined list of set-related fields is included in this chapter of the user guide.

2.1.8 INCLUDE AREA Statement

With the INCLUDE AREA statement, you can select specific areas from the schema or subschema to be represented by the diagram. When you use the INCLUDE AREA statement to specify an area, only the area or areas you select will be represented by the diagram.

2.1.9 LOCATE Statement

With the LOCATE statement, you can position an individual record block relative to another record block in the diagram. When you manually LOCATE record blocks, CA-IDMS/Schema Mapper will automatically position in the diagram any record blocks that you do not manually LOCATE.

2.2 Notations Conventions and Syntax Rules

Be sure to review Exhibit 2.1 for notation conventions and Exhibit 2.2 for parameter syntax rules. Also, review the “Glossary” for the meanings of terms used in this guide.

Example

PROCESS	Keywords appear in UPPERCASE. The minimum required portion of each keyword is UNDERSCORED. If a portion of or an entire keyword is not underscored, you can omit that portion or that keyword.									
,SUBSCHEMA=name	Variables appear in lowercase italics. You must substitute an appropriate value for each variable.									
[,VERSION=number]	Brackets indicate optional clauses.									
PROCESS = <table><tr><td>/</td><td></td><td>\</td></tr><tr><td><</td><td>SCHEMA</td><td>></td></tr><tr><td>\</td><td>SUBSCHEMA</td><td>/</td></tr></table>	/		\	<	SCHEMA	>	\	SUBSCHEMA	/	Braces enclose two or more options. You must select one of them.
/		\								
<	SCHEMA	>								
\	SUBSCHEMA	/								
,field name ...	An ellipsis indicates that a variable number of field names can be specified, separated by commas.									

Exhibit 2.1: Notation Conventions

Item	Explanation
Order of Parameter Statements	<p>CA-IDMS/Schema Mapper executes parameter statements in the order they are supplied. Only the PROCESS statement is mandatory. Enter the parameter statements in this sequence:</p> <p>PROCESS--This mandatory statement must be the first statement entered.</p> <p>OPTIONS--If you want to use OPTIONS parameters enter this statement second.</p> <p>CHARDEF statements follow the OPTIONS statement to control the graphic components of the data structure diagram.</p> <p>If DRECLINE and XRECLINE statements are both used, DRECLINE statements must precede XRECLINE statements.</p> <p>If DSETLINE and XSETLINE statements are both used, DSETLINE statements must precede XSETLINE statements.</p> <p>INCLUDE statements follow OPTIONS, CHARDEF, DRECLINE, XRECLINE, DSETLINE, and XSETLINE statements and precede all LOCATE statements.</p> <p>LOCATE statements are entered last.</p>
Continuing a Parameter Statement	<p>To continue a parameter statement onto the next record, key in a trailing comma. Do not split a keyword phrase between two records. LOCATE statements must be self-contained in individual records.</p>
Entering Blanks In Parameter Statements	<p>You can enter blanks (character spaces) to separate keywords and improve readability in a parameter statement without affecting processing. Do not, however, embed blanks within a keyword or value field.</p>
Entering Parameter Statements	<p>On an 80-character input record, you must enter all parameter syntax between positions 1 and 72, inclusive.</p>

Item	Explanation
Comments	Enter a dollar sign (\$) or an asterisk (*) before a comment, in column 1, to indicate that you want the command copied to the resulting Transfer File. Enter three dollar signs or three asterisks in columns 1-3 to indicate a comment that you do not want copied to the Transfer File (to be discarded by CA-IDMS/Schema Mapper).

Exhibit 2.2: Parameter Syntax Rules

```

PROCESS= / SCHEMA \ [,SCHEMA=name] [,DICTNAME=name]
          < SUBSCHEMA > [,VERSION=number]
          \ LIST / [,SUBSCHEMA=name] [,DICTNODE=name]

OPTIONS [,IXCORNER= / \
              \ 2 > ]
              \ 4 /

          [,DPAGELEN=number] [,DPAGEWID=number]

          [,DPEJECT= / \
              < ON > ] [,BORDER=number]
              \ OFF /

          [,COMPRESS= / \
              < ON > ] [,LINKAGE= / \
              \ OFF / < XREF > ]
                  \ SUPPRESS /

CHARDEF [,RVERCHAR=character] [,RHORCHAR=character]
          [,RULCHAR=character] [,RLLCHAR=character]
          [,RURCHAR=character] [,RLRCHAR=character]
          [,SVERCHAR=character] [,SHORCHAR=character]
          [,SULCHAR=character] [,SLLCHAR=character]
          [,SURCHAR=character] [,SLRCHAR=character]
          [,BRKTCHAR=character] [,LARROW=character]
          [,RARROW=character] [,UARROW=character]
          [,DARROW=character] [,IBAKCHAR=character]
          [,ISLACHAR=character]

[DRECLINE field name,field name...]
[XRECLINE field name,field name...]
[DSETLINE field name,field name...]
[XSETLINE field name,field name...]
[INCLUDE AREA area name...]

[ LOCATE name1 [ [ / \
                  < LEFT > number ] [ < UP > number ] FROM name2 ]
                  \ RIGHT / \ DOWN / ]

```

Exhibit 2.3: CA-IDMS/Schema Mapper Parameter Summary

2.3 PROCESS Statement

A single PROCESS statement is mandatory for each execution of CA-IDMS/Schema Mapper. This statement directs CA-IDMS/Schema Mapper to process a data structure diagram for either a schema or subschema or produce a list of the schemas and subschemas within a dictionary.

To create a data structure diagram with automatic layout and the default format, enter a single PROCESS statement. PROCESS must be the first statement entered.

2.3.1 PROCESS Syntax

```
PROCESS= / SCHEMA \ [,SCHEMA=name] [,DICTNAME=name]
          < SUBSCHEMA > [,VERSION=number]
          \ LIST / [,SUBSCHEMA=name] [,DICTNODE=name]
```

where:

```
PROCESS= / SCHEMA \
          < SUBSCHEMA >
          \ LIST /
```

indicates that you want to execute CA-IDMS/Schema Mapper. Select SCHEMA to create a data structure diagram for a specific schema. Select SUBSCHEMA to create a data structure diagram for a specific subschema. Select LIST to obtain a listing of all schemas and subschemas in a specified dictionary.

[,SCHEMA=name]

specifies the name of the schema to be diagrammed (when PROCESS=SCHEMA) or, when PROCESS=SUBSCHEMA, the schema under which the subschema exists. This parameter is always required to create a data structure diagram.

Default: There is no default. You must supply a valid schema name.

[,VERSION=number]

specifies the version number of the schema. A version number is required when you do not want to use the highest-numbered version.

Default: When PROCESS=SCHEMA, the default is the highest version number available for the schema you select. When PROCESS=SUBSCHEMA, the default is the highest-numbered schema version for the subschema you select.

In the following example, the dictionary contains:

Schema SMSCHM version 3: No subschemas

Schema SMSCHM version 2: Subschema SMSUBS

Schema SMSCHM version 1: Subschema SMSUBS

When PROCESS=SCHEMA,SCHEMA=SMSCHM, CA-IDMS/Schema Mapper uses version 3.

When PROCESS=SUBSCHEMA,SCHEMA=SMSCHM, SUBSCHEMA=SMSUBS, CA-IDMS/Schema Mapper uses version 2.

[,SUBSCHEMA=name]

specifies the name of the subschema to be diagrammed. Supplying a subschema name is necessary only when PROCESS=SUBSCHEMA or the schema contains indexes.

Default: There is no default. You must supply a valid subschema name when the schema contains indexes or PROCESS=SUBSCHEMA.

[,DICTNAME=name]

specifies the name of the dictionary that contains the schema or subschema information. Supplying a dictionary name is necessary only if multiple dictionaries exist in your environment.

Default: CA-IDMS/Schema Mapper uses a standard BIND RUN-UNIT. Therefore, CA-IDMS binds to the default dictionary defined by your installation.

[,DICTNODE=name]

specifies the name of the dictionary node that you want CA-IDMS/Schema Mapper to access. Specifying DICTNODE is necessary only when multiple nodes exist in your environment.

Default: CA-IDMS binds to the default node defined for your installation.

2.4 OPTIONS Statement

The OPTIONS statement allows you to make changes to the layout of the diagram. You can globally specify how indexes are drawn and how much border space is to separate record blocks. You can also change the length and the width of the printed page to match your printer's requirements.

2.4.1 OPTIONS Syntax

```
OPTIONS [,IXCORNER=  $\begin{array}{c} / \quad \backslash \\ < \ 2 \ > \\ \ 4 \ / \end{array}$  ]  
  
          [,DPAGELN=number]      [,DPAGEWID=number]  
  
          [,DPEJECT=  $\begin{array}{c} / \quad \backslash \\ < \text{ON} \ > \\ \text{OFF} \ / \end{array}$  ]      [,BORDER=number]  
  
          [,COMPRESS=  $\begin{array}{c} / \quad \backslash \\ < \text{ON} \ > \\ \text{OFF} \ / \end{array}$  ]      [,LINKAGE=  $\begin{array}{c} / \quad \backslash \\ < \text{XREF} \ > \\ \text{SUPPRESS} \ / \end{array}$  ]
```

where:

OPTIONS

indicates that OPTIONS parameters will be entered. You can enter OPTIONS parameters in any order because each parameter is independent. You can use any or all of the OPTIONS parameters.

Note: Remember to key in a comma between the word OPTIONS and the first parameter. If you omit the comma, CA-IDMS/Schema Mapper generates an error message.

2.4.2 Specifying the Positions of Index

```
[,IXCORNER=  $\begin{array}{c} / \quad \backslash \\ < \ 2 \ > \\ \ 4 \ / \end{array}$  ]
```

(index corner) allows you to control how indexes are drawn in the diagram. An index is represented by a diagonal line that projects from a corner of a record block. Where and how you decide to have indexes drawn is an aesthetic decision.

Default: The default is 2 (opposite diagonal sequence).

2.4.3 Changing Page Dimensions for Paper Size and Printer Flexibility

[,DPAGELEN=number]

(diagram page length) allows you to specify the length of each page in the data structure diagram, in number of lines. DPAGELEN affects only the printing of the diagram; it has no effect on the page length of the Cross-Reference Report or the Audit Report.

This parameter makes CA-IDMS/Schema Mapper adaptable to various paper sizes and printers by allowing you to change how many lines are printed per page. If you supply a new page length value, it must be an integer between 33 and 132, inclusive.

Default: The default page length is 66 lines.

[,DPAGEWID=number]

(diagram page width) allows you to specify the width of each page in the data structure diagram, in number of characters. DPAGEWID affects the printing of only the diagram; it has no effect on the page width of the Cross-Reference Report or the Audit Report.

This parameter makes CA-IDMS/Schema Mapper adaptable to a broad range of paper sizes and printers by allowing you to change the number of characters printed across a page. If you supply a new page width value, it must be an integer between 33 and 132, inclusive.

Default: The default page width is 132 characters.

[,DPEJECT= $\begin{array}{cc} / & \backslash \\ < \text{ON} & > \\ \backslash \text{OFF} & / \end{array}$]

(diagram page eject) allows CA-IDMS/Schema Mapper to issue a page eject after each page in the diagram file when DPEJECT=ON. When DPEJECT=OFF no page eject is issued. DPEJECT affects only the printing of the diagram; it has no effect on the Cross-Reference Report or the Audit Report. ON makes CA-IDMS/Schema Mapper adaptable to use with some laser printers.

Default: The default value is OFF.

Note: DPAGELEN, DPAGEWID, and DPEJECT parameter keywords cannot be abbreviated.

2.4.4 Defining Border Space Between Record Blocks

[, BORDER=number]

globally specifies the number of character spaces around the perimeter of each record block. The minimum border is two character spaces. You cannot specify a border of 0 or 1. A border of 2 means there are at least four character spaces between any two record blocks. A two-character border ensures that CA-IDMS/Schema Mapper has enough room to draw arrows and at least a one-character set connection line.

The maximum border allowed between record blocks is 50 character spaces. A border of 50 means there are at least 100 character spaces between any two record blocks.

Default: The default is 2. This means at least four character spaces always separate a given record block from another record block in the diagram.

2.4.5 Compressing Unused Space in the Data Structure Diagram

[, COMPRESS= $\begin{array}{cc} / & \backslash \\ < \text{ON} & > \\ \backslash \text{OFF} & / \end{array}$]

specifies whether or not you want CA-IDMS/Schema Mapper to remove unused space (created by LOCATE statements) between record blocks in the diagram.

This parameter is valuable when you are using LOCATE statements and executing CA-IDMS/Schema Mapper iteratively to modify the layout of a diagram. See 2.11, “LOCATE Statement” on page 2-47 for a complete discussion. With this parameter, you can wait until a diagram nears completion to compress it.

When COMPRESS=ON, CA-IDMS/Schema Mapper compresses the diagram, eliminating unused space, after all record blocks have been positioned in the diagram.

Default: The default value is ON.

2.4.6 Set Linkage with Areas not Represented in Diagram

[, LINKAGE= $\begin{array}{cc} / & \backslash \\ < \text{XREF} & > \\ \backslash \text{SUPPRESS} & / \end{array}$]

specifies whether or not you want a listing in the Cross-Reference Report (XREF) of sets whose owners and members exist outside of an area specified by the INCLUDE AREA statement. When the INCLUDE AREA statement is used, the area represented in the diagram will contain only those sets that have both owners and members in the area. To prevent the listing from appearing on the Cross-Reference Report, you must use the SUPPRESS option with this parameter.

Default: XREF is the default.

2.5 CHARDEF Statement

CHARDEF is an optional statement with many parameters for tailoring the data structure diagram. If your printer's character set does not include characters that are specified by default, use the CHARDEF statement to substitute print characters that are available to draw the diagram.

For example, in the default format, all arrows pointing to the left are drawn with a less-than sign (<). If this character is not available, you can use the LARROW (left arrow) parameter to specify a character to take its place. You can also change the print characters to achieve a different graphic effect.

2.5.1 CHARDEF Syntax

```
CHARDEF [ ,RVERCHAR=character] [ ,RHORCHAR=character]
        [ ,RULCHAR=character] [ ,RLLCHAR=character]
        [ ,RURCHAR=character] [ ,RLRCHAR=character]
        [ ,SVERCHAR=character] [ ,SHORCHAR=character]
        [ ,SULCHAR=character] [ ,SLLCHAR=character]
        [ ,SURCHAR=character] [ ,SLRCHAR=character]
        [ ,BRKTCHAR=character] [ ,LARROW=character]
        [ ,RARROW=character] [ ,UARROW=character]
        [ ,DARROW=character] [ ,IBAKCHAR=character]
        [ ,ISLACHAR=character]
```

where:

CHARDEF

indicates that CHARDEF parameters will be entered. You can enter CHARDEF parameters in any order because each parameter is independent. You can use any or all of the CHARDEF parameters.

Note: Remember to key in a comma between the word CHARDEF and the first parameter. If you omit the comma, CA-IDMS/Schema Mapper generates an error message.

Do not use a blank space for a character value in the CHARDEF statement. CA-IDMS/Schema Mapper ignores the blank and uses the default.

2.5.2 Specifying Print Characters Used to Draw Record Blocks

[,RVERCHAR=character]

(record vertical character) indicates the character used to draw the vertical lines that make up the left and right sides of the record block.

Default: The default is an asterisk (*).

[,RHORCHAR=character]

(record horizontal character) indicates the character used to draw the horizontal lines that make up the top and bottom of the record block.

Default: The default is an asterisk (*).

[,RULCHAR=character]

(record upper left character) indicates the character used to draw the upper left corner of the record block.

Default: The default is RHORCHAR. The RHORCHAR default is an asterisk (*).

[,RLLCHAR=character]

(record lower left character) indicates the character used to draw the lower left corner of the record block.

Default: The default is RHORCHAR. The RHORCHAR default is an asterisk (*).

[,RURCHAR=character]

(record upper right character) indicates the character used to draw the upper right corner of the record block.

Default: The default is RHORCHAR. The RHORCHAR default is an asterisk (*).

[,RLRCHAR=character]

(record lower right character) indicates the character used to draw the lower right corner of the record block.

Default: The default is RHORCHAR. The RHORCHAR default is an asterisk (*).

2.5.3 Specifying Print Characters Used to Draw Set Connections

[,SVERCHAR=character]

(set vertical character) indicates the character used to draw the vertical lines that show set connections.

Default: The default is an uppercase letter I.

[,SHORCHAR=character]

(set horizontal character) indicates the character used to draw the horizontal lines that show set connections.

Default: The default is a hyphen (-).

[,SULCHAR=character]

(set upper left character) indicates the character used to draw an upper left turn in a set connection.

Default: The default is SHORCHAR. The SHORCHAR default is a hyphen (-).

[,SLLCHAR=character]

(set lower left character) indicates the character used to draw a lower left turn in a set connection.

Default: The default is SHORCHAR. The SHORCHAR default is a hyphen (-).

[,SURCHAR=character]

(set upper right character) indicates the character used to draw an upper right turn in a set connection.

Default: The default is SHORCHAR. The SHORCHAR default is a hyphen (-).

[,SLRCHAR=character]

(set lower right character) indicates the character used to draw a lower right turn in a set connection.

Default: The default is SHORCHAR. The SHORCHAR default is a hyphen (-).

[,BRKTCHAR=character]

(bracket character) indicates the character used to enclose or bracket a set number in a set connection. A bracketed set number is easier to read, especially when the set number contains the number 1 and is part of a vertical set connection line.

Default: The default is the number symbol (#).

2.5.4 Specifying Print Characters Used to Draw Arrows

[,LARROW=character]

(left arrow) indicates the character used to print the left arrows that show set relationships. If the default character is not available, the most likely graphic substitute is an uppercase letter C. You can, however, use any other character as a substitute.

Default: The default is a less-than sign (<).

[,RARROW=character]

(right arrow) indicates the character used to print the right arrows that show set relationships. If the default character is not available, the most likely graphic substitute is an uppercase letter D. You can, however, use any other character as a substitute.

Default: The default is a greater-than sign (>).

[,U~~A~~RROW=character]

(up arrow) indicates the character used to print a set connection arrow pointing up toward another record block.

Default: The default is an uppercase letter A.

[,D~~A~~RROW=character]

(down arrow) indicates the character used to print a set connection arrow pointing down toward another record block.

Default: The default is an uppercase letter V.

2.5.5 Specifying Print Characters Used to Draw Index Lines

[,I~~S~~LACHAR=character]

(index slash character) indicates the character used to draw indexes that project diagonally from the upper right corners and lower left corners of record blocks.

Default: The default is a slash (/).

[,I~~B~~A~~K~~CHAR=character]

(index backslash character) indicates the character used to draw the index lines that project diagonally from the upper left corners and lower right corners of record blocks when IXCORNER=4.

Default: The default is an uppercase letter X.

2.6 DRECLINE Statement

DRECLINE (diagram record line) allows you to globally specify the format (text) of the record block in the data structure diagram. The format of a record block is defined as one or more lines containing one or more record definition field names.

The format of the record block in the data structure diagram is controlled by default unless you use the DRECLINE statement. Exhibit 2.4 shows an example of a record block in the four line default format.

Exhibit 2.5 at the end of this section, lists all possible field names that you can enter in a DRECLINE statement, including the CA-IDMS/DB ANALYZER statistics field names and their descriptions.

Note: DRECLINE is a global parameter statement. Once specified, it modifies every record block in the data structure diagram. For compatibility with earlier versions of CA-IDMS/Schema Mapper, any RECLINE statements in the input file are treated as DRECLINE statements, and CA-IDMS/Schema Mapper issues a warning message.

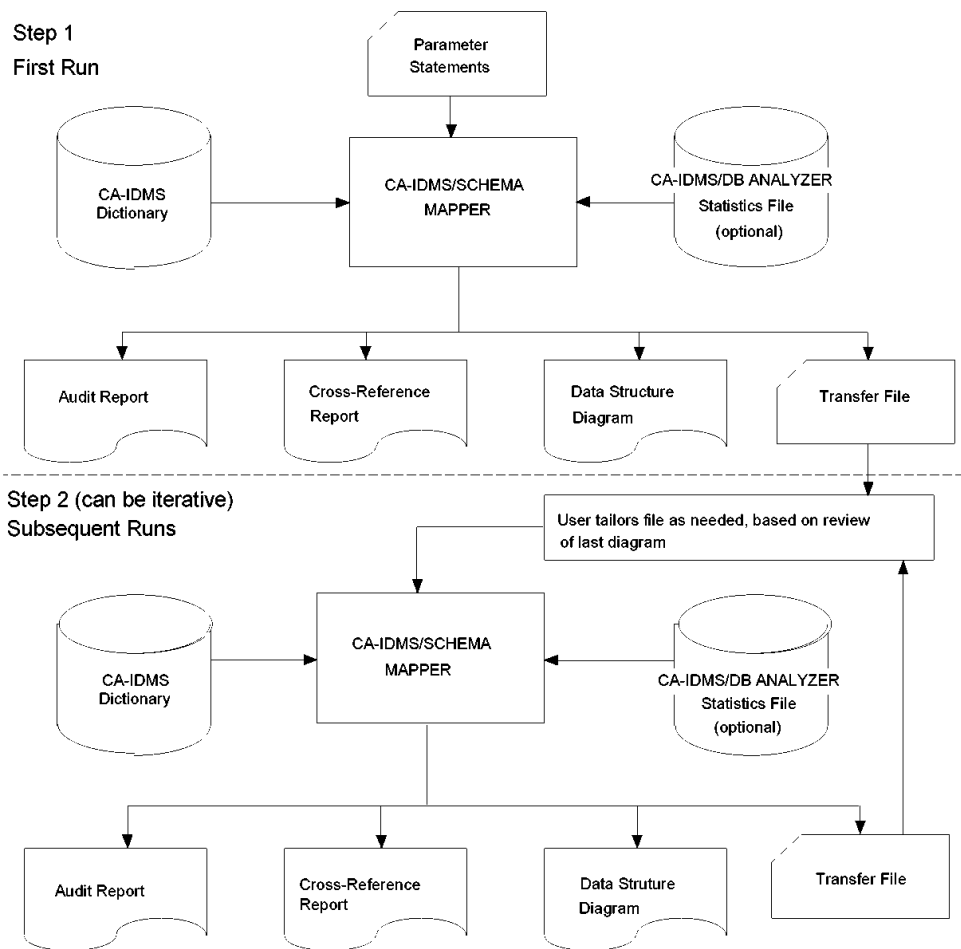


Exhibit 2.4: Default Record Block with Field Names in a Four Line Format

2.6.1 DRECLINE Syntax

[DRECLINE field name,field name...]

where:

field name,field name... — indicates the field name (or sequence of field names) to be included in the record block in the diagram. When two or more field names are used per line, they must be separated by commas. See Exhibit 2.5 for a list of field names.

Rules:

- DRECLINE statements must follow PROCESS, OPTIONS and CHARDEF statements and must precede XRECLINE, DSETLINE, XSETLINE, INCLUDE AREA, and LOCATE statements.

- The maximum number of characters allowed for each line of a record block is 120.
- Enter DRECLINE field names in the order in which you want them to appear in the record block. Separate field names by a comma.
- You cannot use a field name more than once (except for BLANK, LABEL, or LOC-CTRL with unique subscripts).
- One DRECLINE statement cancels all record definition (text) defaults. When you use DRECLINE, you must specify the entire record block format in the data structure diagram from scratch.

Default: Exhibit 2.4 describes the default, if no DRECLINE statement is specified.

2.6.2 How to Use the DRECLINE Statement

A record block is made up of one or more lines of text that contain one or more fields. When you want to globally redefine the format of the record block in the data structure diagram, write one DRECLINE statement for each line of data in the record block. List the field names in the order in which you want the fields to appear in the record blocks.

2.6.3 Graphic Considerations

The size and the shape of the record block affects the overall appearance of the diagram. When you use DRECLINE to redesign the record block, be aware that the size and the shape of the record block affects the overall appearance of the diagram. For example, if you enter all field names in one DRECLINE statement, the resulting record block is long and thin and probably unacceptable. Keep in mind that the width of the default record block is 32 characters.

CA-IDMS/Schema Mapper adjusts record block size to accommodate the longest line. You can place any number of field names on a line. CA-IDMS/Schema Mapper adjusts the width of the record block to accommodate the longest line. However, a maximum of 120 characters can be entered per line.

CA-IDMS/Schema Mapper centers field values within the number of character spaces for that field. In addition to centering field values within the number of character spaces for that field, CA-IDMS/Schema Mapper centers fields across a line, based on the total character length for the longest line in the record block.

2.6.4 Creating Blank Fields and Blank Lines In Record Blocks

You can insert blank fields and blank lines in the record block to make room for handwritten notes or for graphic appeal.

To Create A Blank Field Inside A Record Block:

Enter BLANK *nn* as a field name, where *nn* specifies the blank field length in character spaces. The value *nn* must be an integer between 1 and 32, inclusive. A group of DRECLINE statements may contain a maximum of 20 BLANK fields.

To Create A Blank Line Inside A Record Block:

Use DRECLINE by itself (without a field name). The length of the blank line will match the length of the longest line in the record block.

2.6.5 Placement of Multiple CALC Keys

You can use an optional subscript with the LOC-CTRL field to specify placement of multiple CALC key names within the record block in the data structure diagram. The syntax is:

LOC-CTRL(ss)

where *ss* may be an integer from 1 to 256 or the capital letter N.

When you use an integer from 1 to 256, you are explicitly placing the CALC keys you want to see at the positions you specify. For example, LOC-CTRL(1) will position the first CALC key found in the dictionary for each record type. LOC-CTRL(2) will position the second CALC key, etc. If you use LOC-CTRL without a subscript, it is equivalent to LOC-CTRL(1).

When you use only the letter N, CA-IDMS/Schema Mapper will place all CALC keys in the record block in the diagram beginning at the position indicated by LOC-CTRL(N). The first CALC key will appear at that position followed by the remaining CALC keys one per line.

You can choose to explicitly place one or more CALC keys and have CA-IDMS/Schema Mapper implicitly place the remaining ones. Any CALC keys that have not been positioned by an explicit subscript will appear one per line, starting with the position indicated by LOC-CTRL(N). See 2.6.6, “Sample DRECLINE Statements and Resulting Record Blocks” on page 2-23.

When you do not use LOC-CTRL(N) and there are more CALC keys than you have explicitly positioned, the additional CALC keys will not appear in the record block. An informative message will appear in the Audit Report to let you know this has happened. For example, if one of your record types has five CALC keys, but you only use LOC-CTRL(1), LOC-CTRL(2), and LOC-CTRL(3) in your DRECLINE statements, then the fourth and fifth CALC keys will not appear in the record block in the data structure diagram.

Also, LOC-CTRL(N) affects only CALC keys that follow the highest explicit position. If any lower subscripts are omitted from the DRECLINE statements, the corresponding CALC keys will not appear in the diagram. For example, if a record has seven CALC keys and the DRECLINE statements use LOC-CTRL(1), LOC-CTRL(4), and LOC-CTRL(N), then LOC-CTRL(N) positions CALC keys 5, 6, and 7 only. CALC keys 2 and 3 will not appear in the diagram. Any CALC key that does not appear in

the diagram, for any reason, will be listed in the Audit Report with an informative message.

CA-IDMS/Schema Mapper makes all record blocks the same size (remember, DRECLINE is a global parameter). Therefore, if only one of your record types contains multiple CALC keys and you want all the keys for that record type to appear in the record block, then every record block in the diagram will contain space for multiple CALC keys. This is true even if a particular record type does not have a location mode of CALC.

By default, CA-IDMS/Schema Mapper uses LOC-CTRL(N). Therefore, all default record blocks contain space for all CALC keys.

2.6.6 Sample DRECLINE Statements and Resulting Record Blocks

To Create A Three-Line Record Block:

To create a three-line record block that has the record name on line 1, the area and location mode on line 2, and the record size and description on line 3, use these statements:

```
DRECLINE REC-NAME
DRECLINE AREA,LOC-MODE
DRECLINE LENGTH,REC-DESC
```

A resulting record block looks like this:

```
*****
*           SPORTS                               *
*           STUDENT-REGION           CALC         *
*  32    NON-COMPETITIVE AND COMPETITIVE SPORTS *
*****
```

To Create A Six-Line Record Block:

To create a six-line record block that has the record name on line 1, a blank second line, the area, location mode, and record size on line 3, the description on line 4, the duplicates option on line 5, and the location control on line 6, use these statements:

```
DRECLINE REC-NAME
DRECLINE
DRECLINE AREA,LOC-MODE,LENGTH
DRECLINE REC-DESC
DRECLINE DUPS-OPT
DRECLINE LOC-CTRL
```

Here is a resulting record block:

```
*****
*           SPORTS                               *
*           STUDENT-REGION           CALC         32 *
*  NON-COMPETITIVE AND COMPETITIVE SPORTS         *
*           DN                                     *
*           SPORT-NAME                           *
*****
```

To Create Blank Fields or Blank Lines for Handwritten Notes or Graphic Appeal:

A record block with a blank second line and a blank, 10-character field in line 3 is specified like this:

```
DRECLINE REC-NAME
DRECLINE
DRECLINE AREA,BLANK 10,LOC-MODE
DRECLINE LENGTH,REC-DESC
```

Note: You must put a space between the word BLANK and the blank field length. Leaving out the space results in an error message.

Here is a record block with the above format:

```
*****
*           SPORTS                               *
*                                           *
*           STUDENT-REGION                      CALC *
*  32      NON-COMPETITIVE AND COMPETITIVE SPORTS *
*****
```

To Combine Explicit and Implicit Placement of CALC Keys:

A record block with the first CALC key on the second line and additional CALC keys on the last lines is specified like this:

```
DRECLINE REC-NAME,LOC-MODE
DRECLINE LOC-CTRL(1)
DRECLINE AREA
DRECLINE LOC-CTRL(N)
```

Here is a resulting record block:

```
*****
*           STUDENT                      CALC      *
*           STUD-ID                      *
*           STUDENT-REGION                *
*           STUD-LEVEL                    *
*           STUD-PA                       *
*           STUD-MA                       *
*****
```

Use of LOC-CTRL(N) always reserves a position in the record block, even when there are not enough CALC keys to actually need it.

Field Names for the DRECLINE and XRECLINE Statements:

Field Name	Size (# of chars)	Description
REC-NAME (record name)	16	The name of the record type.
LOC-MODE	6	How records of this type are physically located in their areas: CALC, VIA, or DIRECT
AREA	16	The name of the area in which the record type is located.
LENGTH (record size)	6	The actual data length of the record (if fixed length) or the maximum data length of the record (if variable length), in bytes. If the value obtained from the dictionary is less than 1, CA-IDMS/Schema Mapper uses a value of 1 in the record block. This happens, for example, when a schema record is defined without any elements and is consistent with the value reported by IDMSRPTS or RECDES.
LENGTH-MODE	2	The four record modes: F (fixed), V (variable), FC (fixed compressed), or VC (variable compressed).
DUPS-OPT (Duplicates Option)	2	For CALC records, the disposition of records with duplicate control keys: DN (duplicates not allowed), DF (duplicates first), or DL (duplicates last). This field is blank for non-CALC records.
REC-DESC (Record Description)	40	The record description field from the dictionary. Provides an alternate (and more descriptive) record name.
LOC-CTRL (Location Control)	32	CALC key name or VIA set name. This is always DIRECT-DBK for DIRECT records.
REC-ID (Record Identification)	4	Unique identification number of record type.
BLANK nn	nn	Name used to specify a blank field.
UN-NUM-OCC	14	The total number of record occurrences for this record type.
UN-OCC-LEN-CAVG	17	The average compress occurrence length for this record type.

Field Name	Size (# of chars)	Description
UN-OCC-LEN-CMAX	14	The maximum compress occurrence length for this record type.
UN-OCC-LEN-CMIN	14	The minimum compress occurrence length for this record type.
UN-OCC-LEN-DAVG	17	The average decompress occurrence length for this record type.
UN-OCC-LEN-DMAX	14	The maximum decompress occurrence length for this record type.
UN-OCC-LEN-DMIN	14	The minimum decompress occurrence length for this record type.
LABEL 'any string' -or- LABEL "any string"	nn	Used to put labels in a record description.

Exhibit 2.5: Field Names for the DRECLINE and XRECLINE Statements

2.7 XRECLINE Statement

XRECLINE (cross-reference record line) allows you to include and format record block information on the Cross-Reference Report. The record description on the report contains the same information as a record block in the data structure diagram. If you do not use the XRECLINE statement, CA-IDMS/Schema Mapper provides only the record name and location in the Cross-Reference Report. The reports in Exhibit 2.6 show a record description before and after the addition of XRECLINE statements.

The XRECLINE statement defines the format of the record description line by line, listing the field names, blank fields, labels, and blank lines to be included in the record description. See Exhibit 2.5 for a list of all possible field names that you can enter in an XRECLINE statement.

Note: XRECLINE is a global parameter statement. Once specified, it modifies every record description in the Cross-Reference Report.

2.7.1 XRECLINE Syntax

[XRECLINE field name,field name...]

where:

field name,field name... — indicates the field name (or sequence of field names) to be included in the record description in the Cross-Reference Report.

Rules:

- XRECLINE statements must follow PROCESS, OPTIONS, CHARDEF, and DRECLINE statements and must precede DSETLINE, XSETLINE, INCLUDE AREA, and LOCATE statements.
- The maximum number of characters allowed for each line of a record description is 120.
- Enter XRECLINE field names in the order in which you want them to appear in the Cross-Reference Report. Separate field names by a comma.
- You cannot use a field name more than once (except for BLANK, LABEL, and LOC-CTRL with unique subscripts).

Default: The default XRECLINE value is XRECLINE REC-NAME.

2.7 XRECLINE Statement

CA-IDMS/Schema Mapper: mm/dd/yy hh:mm:ss SUBSCHEMA DBRR1016, SCHEMA DSRR1016, VERSION 1, DICTNAME=TKIT, DBNAME=			
RECORD CROSS-REFERENCE LISTING			
RECORD NAME	LOCATION		/ PAGE
-----	-----		-----
ACTIVITY	94,	27	/ AA
ARTS	150,	27	/ BA
CLASS	94,	74	/ AB
CON-TROL	1,	12	/ AA
DEPT	39,	48	/ AA
DREPORT	39,	12	/ AA
GREPORT	39,	27	/ AA
MUSIC	150,	48	/ BA
PERIOD	94,	95	/ AB
PREREQ	39,	95	/ AB
ROOM	150,	95	/ BB
SCHEDULE	150,	12	/ BA
SPORTS	150,	74	/ BB
STUDENT	94,	12	/ AA
SUBJECT	39,	74	/ AB
TEACHER	94,	48	/ AA
CA-IDMS/Schema Mapper: mm/dd/yy hh:mm:ss SUBSCHEMA DBRR1016, SCHEMA DSRR1016, VERSION 1, DICTNAME=TKIT, DBNAME=			
RECORD CROSS-REFERENCE LISTING			
RECORD NAME	LOCATION		/ PAGE
-----	-----		-----
ACTIVITY	94,	27	/ AA
ACTIVITY			
CLASS-REGION	CALC	32	
DN			
ACT-CODE			
ARTS	150,	27	/ BA
ARTS			
CLASS-REGION	VIA	32	
ACTIVITY-TYPES			
CLASS			
CLASS	94,	74	/ AB
CLASS-REGION	VIA	12	
SUBJECT-CLASS			
CON-TROL	1,	12	/ AA
CON-TROL			
LOC-REGION	DIRECT	32	
DIRECT-DBK			
DEPT	39,	48	/ AA
DEPT			
DEPT-REGION	CALC	52	
DN			
DEPT-NAME			
DREPORT	39,	12	/ AA
DREPORT			
STUDENT-REGION	VIA	68	
STUDENT-REPORTS			

Exhibit 2.6: Sample Cross-Reference Reports Before and After Using XRECLINE Statement

2.7.2 How to Use the XRECLINE Statement

A record description is made up of one or more lines of text that contain one or more fields. When you want to globally redefine the format of the record description in the Cross-Reference Report, write one XRECLINE statement for each line of data in the record description.

From Exhibit 2.5, select the field names you want to include in the record description. Enter these field names in the XRECLINE statement in the order in which you want them to appear on the Cross-Reference Report.

2.7.3 Creating Blank Fields and Blank Lines In Record Descriptions

You can insert blank fields and blank lines in the record description for readability and to make room for handwritten notes.

To Create A Blank Field:

Enter BLANK *nn* as a field name, where *nn* specifies the length of the blank field in character spaces. The value *nn* must be an integer between 1 and 32, inclusive. A group of XRECLINE statements may contain a maximum of 20 BLANK fields.

To Create A Blank Line:

Use XRECLINE by itself (without a field name). The length of the blank line matches the length of the longest line in the record description.

2.7.4 Graphic Considerations

CA-IDMS/Schema Mapper adjusts the record description size to accommodate the longest line. It is possible to place all available record descriptions on one line; CA-IDMS/Schema Mapper adjusts the size of the record field to accommodate the line. A maximum of 120 characters can be entered per line.

All record text in the Cross-Reference Report is left justified, based on record description sizes. The record text for each field is left justified within the maximum number of characters defined for that field.

2.7.5 Placement of Multiple CALC Keys

You can use an optional subscript with the LOC-CTRL field to specify placement of multiple CALC key names within the record description in the Cross-Reference Report. The syntax is:

LOC-CTRL (ss)

where *ss* may be an integer from 1 to 256 or the capital letter N.

When you use an integer from 1 to 256, you are explicitly placing the CALC keys you want to see at the positions you specify. For example, LOC-CTRL(1) positions the first CALC key found in the dictionary for each record type. LOC-CTRL(2) positions the second CALC key, etc. If you use LOC-CTRL without a subscript, it is equivalent to LOC-CTRL(1).

When you use only the letter N, CA-IDMS/Schema Mapper places all CALC keys in the record description in the Cross-Reference Report beginning at the position indicated by LOC-CTRL(N). The first CALC key appears at that position followed by the remaining CALC keys one per line.

You can choose to explicitly place one or more CALC keys and have CA-IDMS/Schema Mapper implicitly place the remaining ones. Any CALC keys that have not been positioned by an explicit subscript appear one per line, starting with the position indicated by LOC-CTRL(N).

When you do not use LOC-CTRL(N) and there are more CALC keys than you have explicitly positioned, the additional CALC keys do not appear in the record description. An informative message appears in the Audit Report to let you know this has happened. For example, if one of your record types has five CALC keys, but you only use LOC-CTRL(1), LOC-CTRL(2), and LOC-CTRL(3) in your XRECLINE statements, then the fourth and fifth CALC keys do not appear in the record description in the Cross-Reference Report.

Also, LOC-CTRL(N) affects only CALC keys that follow the highest explicit position. If any lower subscripts are omitted from the XRECLINE statements, the corresponding CALC keys do not appear in the Cross-Reference Report. For example, if a record has seven CALC keys and the XRECLINE statements use LOC-CTRL(1), LOC-CTRL(4), and LOC-CTRL(N), then LOC-CTRL(N) positions CALC keys 5, 6, and 7 only. CALC keys 2 and 3 do not appear in the Cross-Reference Report. Any CALC key that does not appear in the Cross-Reference Report, for any reason, are listed in the Audit Report with an informative message.

CA-IDMS/Schema Mapper makes all record descriptions the same size (remember, XRECLINE is a global parameter). Therefore, if only one of your record types contains multiple CALC keys and you want all the keys for that record type to appear in the record description, then every record description in the Cross-Reference Report contains space for multiple CALC keys. This is true even if a particular record type does not have a location mode of CALC.

2.7.6 Sample XRECLINE Statements and Resulting Record Descriptions

To Create A Three-Line Record Description:

To create a three-line record description that has the record name on line 1, the area and location mode on line 2, and the record size and description on line 3, use these statements:

```
XRECLINE REC-NAME
XRECLINE AREA,LOC-MODE
XRECLINE LENGTH,REC-DESC
```

A resulting record description looks like this:

```
ACTIVITY                                94,    27 / AA
ACTIVITY
CLASS-REGION                CALC
32
```

To Create A Six-Line Record Description:

To create a six-line record description that has the record name on line 1, a blank second line, the area, location mode, and record size on line 3, the description on line 4, the duplicates option on line 5, and the location control on line 6, use these statements:

```
XRECLINE REC-NAME
XRECLINE
XRECLINE AREA,LOC-MODE,LENGTH
XRECLINE REC-DESC
XRECLINE DUPS-OPT
XRECLINE LOC-CTRL
```

Here is a resulting record description:

```
ACTIVITY                                94,    27 / AA
ACTIVITY
CLASS-REGION                CALC        32
DN
ACT-CODE
```

To Create Blank Fields or Blank Lines for a record description:

A record description with a blank second line and a blank, 10-character field in line 3 is specified like this:

```
XRECLINE REC-NAME
XRECLINE
XRECLINE AREA,BLANK 10,LOC-MODE
XRECLINE LENGTH,REC-DESC
```

Note: You must put a space between the word BLANK and the blank field length. Leaving out the space results in an error message.

Record description with the above format:

```
ACTIVITY                                94,    27 / AA
ACTIVITY
CLASS-REGION                CALC
32
```

To Combine Explicit and Implicit Placement of CALC Keys:

A record description with the first CALC key on the second line and additional CALC keys on the last lines is specified like this:

```
XRECLINE REC-NAME,LOC-MODE
XRECLINE LOC-CTRL(1)
XRECLINE AREA
XRECLINE LOC-CTRL(N)
```

Resulting record description:

```
ACTIVITY                                94,    27 / AA
  ACTIVITY                                CALC
  ACT-CODE
  CLASS-REGION
```

Use of LOC-CTRL(N) always reserves a position in the record description, even when there are not enough CALC keys to actually need it.

2.8 DSETLINE Statement

DSETLINE (diagram set line) allows you to include set descriptions in the data structure diagram and specify the format of these descriptions. The DSETLINE statement defines the format of the set description line by line, listing the field names, blank fields, labels, and blank lines to be included.

A set description consists of the set name, membership options, pointer options, and set order. You can quickly find record blocks and set fields in the data structure diagram by looking at the cross-reference of their locations in the Cross-Reference Report.

Exhibit 2.8 lists all possible field names that you can enter in the DSETLINE statement, including the CA-IDMS/DB ANALYZER statistics field names and their descriptions.

Note: DSETLINE is a global parameter statement. Once specified, it adds a description of every set in the diagram.

If the DSETLINE statement is not included, CA-IDMS/Schema Mapper processes the data structure diagram without set information.

2.8.1 DSETLINE Syntax

[DSETLINE field name,field name...]

where:

field name,field name... — indicates the field name (or series of field names) to be included in the set description in the data structure diagram.

Rules:

- DSETLINE statements must follow PROCESS, OPTIONS, CHARDEF, DRECLINE, and XRECLINE statements and precede XSETLINE, INCLUDE AREA, and LOCATE statements.
- The maximum number of characters allowed for each line of a set description is 120.
- Enter DSETLINE field names in the order in which you want them to appear in the diagram. When two or more field names are used per line, they must be separated by a comma.
- You cannot use a field name more than once (except for BLANK, LABEL or SET-ORDER with unique subscripts).

For compatibility with earlier versions, any SETLINE statements in the input file are treated as XSETLINE statements and CA-IDMS/Schema Mapper issues a warning message.

Default: There is no default DSETLINE value.

2.8.2 How to Use the DSETLINE Statement

A set description is made up of one or more lines of text that contain one or more fields. You must supply one DSETLINE statement for each line of text you want included in the set description in the data structure diagram.

Select the field names you want included in the set description from the list of possible field names in Exhibit 2.8. Enter the field names in the order in which you want the fields to appear in the set description.

2.8.3 Creating Blank Fields and Blank Lines in Set Descriptions

You can insert blank fields or blank lines in the set description to make room for handwritten notes or for graphic appeal.

To Create A Blank Field Within A Set Description:

Enter BLANK *nn* as a field name, where *nn* specifies the blank field length in character spaces. The value *nn* must be an integer between 1 and 32, inclusive. A group of DSETLINE statements may contain a maximum of 20 BLANK fields.

To Create A Blank Line Within A Set Description:

Use DSETLINE by itself (without a field name).

2.8.4 Graphic Considerations

The size and the shape of the set field description affects the overall appearance of the data structure diagram. When you use DSETLINE to redesign the set field description, be aware that the size and the shape of the set description affects the overall appearance of the data structure diagram. For example, if you enter all set field names in one DSETLINE statement, the resulting set field description is long and thin and probably unacceptable.

CA-IDMS/Schema Mapper adjusts set field description size to accommodate the longest line. You can place any number of set field names on a line.

CA-IDMS/Schema Mapper adjusts the width of the set field description to accommodate the longest line. However, a maximum of 120 characters can be entered per line.

CA-IDMS/Schema Mapper centers field values within the number of character spaces for that field. In addition to centering field values within the number of character spaces for that field, CA-IDMS/Schema Mapper centers set fields across a line, based on the total character length for the longest line in the set field description.

2.8.5 Placement of Multiple Sort or Index Keys

You can use an optional subscript with the SET-ORDER field to specify placement of multiple sort or index key names within the set description.

The syntax is:

SET-ORDER(ss)

where *ss* may be an integer from 1 to 256 or the capital letter N.

When you use an integer from 1 to 256, you are explicitly placing the sort or index keys you want to see at the positions you specify. For example, SET-ORDER(1) positions the first sort or index key found in the dictionary for each set or index. SET-ORDER(2) positions the second sort or index key, etc. If you use SET-ORDER without a subscript, it is equivalent to SET-ORDER(1).

When you use only the letter N, CA-IDMS/Schema Mapper places all sort or index keys in the set description beginning at the position indicated by SET-ORDER(N). The first sort or index key appears at that position followed by the remaining sort or index keys one per line.

You can choose to explicitly place one or more sort or index keys and have CA-IDMS/Schema Mapper implicitly place the remaining ones. Any sort or index keys that have not been positioned by an explicit subscript appear one per line, starting with the position indicated by SET-ORDER(N). See 2.8.6, “Sample DSETLINE Statements and Resulting Set Descriptions” on page 2-36.

When you do not use SET-ORDER(N) and there are more sort or index keys than you have explicitly positioned, the additional sort or index keys do not appear in the set description. An informative message appears in the Audit Report to let you know this has happened. For example, if one of your sets or indexes has five sort or index keys, but you only use SET-ORDER(1), SET-ORDER(2), and SET-ORDER(3) in your DSETLINE statements, then the fourth and fifth sort or index keys do not appear in the set description.

Also, SET-ORDER(N) affects only sort or index keys that follow the highest explicit position. If any lower subscripts are omitted from the DSETLINE statements, the corresponding sort or index keys do not appear in the data structure diagram. For example, if a set has five sort keys and the DSETLINE statements use SET-ORDER(1), SET-ORDER(3), and SET-ORDER(N), then SET-ORDER(N) positions sort keys 4 and 5 only. Sort key 2 does not appear in the data structure diagram. Any sort or index key that does not appear in the data structure diagram, for any reason, is listed in the Audit Report with an informative message.

CA-IDMS/Schema Mapper makes all set descriptions the same size (remember, DSETLINE is a global parameter). Therefore, when only one of your sets or indexes contains multiple sort or index keys and you want all the keys for that set or index to appear in the set description, then every set description in the data structure diagram

contains space for multiple sort or index keys. This is true even if a particular set does not have a set order of sorted.

2.8.6 Sample DSETLINE Statements and Resulting Set Descriptions

To Create a Three-Line Set Description:

To create a three-line set description containing the set name in line 1, the set order in line 2, and the pointers in line 3, use these parameter statements:

```
DSETLINE SET-NAME
DSETLINE SET-ORDER
DSETLINE POINTERS
```

A resulting set description looks like this:

```

*****
#1#->*          SPORTS          *
ACTIVITY-TYPES * 8112  F      32    CALC  DN *
LAST           *          SPORT-NAME      *
NPO            *          STUDENT-REGION   *
*****
```

To Create Blank Fields or Blank Lines in a Set Description:

You can create blank fields or blank lines in the set description to make room in the data structure diagram for handwritten information. A set description with a blank five-character field in the first line and a blank second line is specified like this:

```
DSETLINE SET-NAME,BLANK 5,SET-ORDER
DSETLINE
```

The resulting set description:

```

*****
#1#->*          SPORTS          *
ACTIVITY-TYPES      LAST      * 8112  F      32    CALC  DN *
                      *          SPORT-NAME      *
                      *          STUDENT-REGION   *
*****
```

To Combine Explicit and Implicit Placement of Sort Keys:

A set description with the first sort key on the second line and additional sort keys on the last lines is specified like this:

```
DSETLINE SET-NAME
DSETLINE SET-ORDER(1)
DSETLINE POINTERS,MEM-OPT,SET-MODE
DSETLINE SET-ORDER(N)
```

The resulting set description:


```

/
#
6
#
/
*****
DEPT-TEACHER
LAST
NPO OM
*****
*      DEPT      *
* 1000  F      52  CALC  DN  *--#5#----->*      TEACHER      *
*      DEPT-NAME  *--#9#----->*      TCHR-ID      *
*      DEPT-REGION *      X-TCHR-LNAME *      DEPT-REGION *
*****
ASC TCHR-LNAME-A DL *****
N  MA MODE = INDEX

```

Use of SET-ORDER(N) always reserves a position in the set description, even when there are not enough sort or index keys to actually need it.

2.9 XSETLINE Statement

XSETLINE (cross-reference set line) allows you to globally specify the format of the set description data in the Cross-Reference Report. The XSETLINE statement works in much the same way as XRECLINE.

Set descriptions contain identification and descriptive information about the set, including set name, membership options, pointer options, and set order. You can quickly find record blocks and sets in the data structure diagram by looking at the cross-reference of their locations in the Cross-Reference Report.

Exhibit 2.7 shows sample Cross-Reference Reports before and after using the XSETLINE statement. You can also exclude set fields from the Cross-Reference Report. If XSETLINE is not specified, the report provides only set names, modes, pointers, and membership options. Exhibit lists all possible field names that you can enter in the XSETLINE statement.

Note: XSETLINE is a global parameter statement. You cannot modify the format of an individual set description in the Cross-Reference Report. Using one XSETLINE statement cancels all set-related defaults. Specify all of the field names you want included in the Cross-Reference Report.

2.9.1 XSETLINE Syntax

[XSETLINE field name,field name...]

where:

field name,field name... — indicates the field name (or series of field names) to be included in the set description. See Table 2.2 for a complete list of possible field names.

Rules:

- XSETLINE statements must follow PROCESS, OPTIONS, CHARDEF, DRECLINE, XRECLINE and DSETLINE statements and must precede INCLUDE AREA and LOCATE statements.
- The maximum number of characters allowed for each line of a set description is 120.
- Enter XSETLINE field names in the order in which you want them to appear in the Cross-Reference Report. When two or more field names are used per line, they must be separated by a comma.
- You cannot use a field name more than once (except for BLANK or SET-ORDER with unique subscripts).

For compatibility with earlier versions, any SETLINE statements appearing in the input file are treated as XSETLINE statements, and CA-IDMS/Schema Mapper issues a warning message.

Default: The default XSETLINE values are:

```
XSETLINE SET-NAME
XSETLINE POINTERS, MEM-OPT
XSETLINE SET-ORDER
```

CA-IDMS/Schema Mapper: mm/dd/yy hh:mm:ss SUBSCHEMA DBRR1016, SCHEMA DSRR1016, VERSION 1, DICTNAME=TKIT, DBNAME=

SET/INDEX CROSS-REFERENCE LISTING

SET NUMBER	SET OR INDEX NAME	OWNER	LOCATION	/	PAGE	MEMBER	LOCATION	/	PAGE
1	ACTIVITY-TYPES	ACTIVITY	94,	27	/ AA	ARTS	150,	27	/ BA
						ACTIVITY-TYPES			
						NP OA			
						LAST			
						MUSIC	150,	48	/ BA
						ACTIVITY-TYPES			
						NP OA			
						LAST			
						SPORTS	150,	74	/ BB
						ACTIVITY-TYPES			
						NPO OM			
						LAST			
2	CLASS-SCHEDULE	CLASS	94,	74	/ AB	SCHEDULE	150,	12	/ BA
						CLASS-SCHEDULE			
						NPO MA			
						ASC ROSTER-NO DL			

CA-IDMS/Schema Mapper: mm/dd/yy hh:mm:ss SUBSCHEMA DBRR1016, SCHEMA DSRR1016, VERSION 1, DICTNAME=TKIT, DBNAME=

SET/INDEX CROSS-REFERENCE LISTING

SET NUMBER	SET OR INDEX NAME	OWNER	LOCATION	/	PAGE	MEMBER	LOCATION	/	PAGE
1	ACTIVITY-TYPES	ACTIVITY	94,	27	/ AA	ARTS	150,	27	/ BA
						ACTIVITY-TYPES			
						NP OA			
						LAST			
						MUSIC	150,	48	/ BA
						ACTIVITY-TYPES			
						NP OA			
						LAST			
						SPORTS	150,	74	/ BB
						ACTIVITY-TYPES			
						NPO OM			
						LAST			
2	CLASS-SCHEDULE	CLASS	94,	74	/ AB	SCHEDULE	150,	12	/ BA
						CLASS-SCHEDULE			
						NPO MA			
						ASC ROSTER-NO DL			
3	DEPT-ACADEMIC	DEPT	39,	48,	/ AA	SUBJECT	39,	74	/ AB
						DEPT-ACADEMIC			
						NO MM			
						NEXT			
4	DEPT-GENERAL	DEPT	39,	48	/ AA	SUBJECT	39,	74	/ AB
						DEPT-GENERAL			
						NO MM			
						NEXT			
5	DEPT-TEACHER	DEPT	39,	48	/ AA	TEACHER	94,	48	/ AA
						DEPT-TEACHER			
						NPO OM			
						LAST			

Exhibit 2.7: Sample Cross-Reference Reports Before and After Using XSETLINE Statement

2.9.2 How to Use the XSETLINE Statement

A set description is made up of one or more lines of text that contain one or more fields. You must supply one XSETLINE statement for each line of text you want included in the set description.

To redesign the format of the set description in the Cross-Reference Report, select the field names you want included in the set description format from Exhibit 2.8. Enter the field names in the order in which you want the fields to appear in the set description.

2.9.3 Creating Blank Fields and Blank Lines in Set Descriptions

You can insert blank fields or blank lines in the set description to make room for handwritten notes or for readability.

To Create A Blank Field Within A Set Description:

Enter BLANK *nn* as a field name, where *nn* specifies the blank field length in character spaces. The value *nn* must be an integer between 1 and 32, inclusive. A group of XSETLINE statements may contain a maximum of 20 BLANK fields.

To Create A Blank Line Within A Set Description:

Use XSETLINE by itself (without a field name).

2.9.4 Graphic Considerations

CA-IDMS/Schema Mapper adjusts the set description size to accommodate the longest line. It is possible to place all available set description field names on one line; CA-IDMS/Schema Mapper adjusts the size of the set description to accommodate the line. A maximum of 120 characters can be entered per line.

All set text in the Cross-Reference Report is left justified, based on field sizes. The set text for each field is left justified within the maximum number of characters defined for that field.

2.9.5 Placement of Multiple Sort or Index Keys

You can use an optional subscript with the SET-ORDER field to specify placement of multiple sort or index key names within the set description.

The syntax is:

SET-ORDER (ss)

where *ss* may be an integer from 1 to 256 or the capital letter N.

When you use an integer from 1 to 256, you are explicitly placing the sort or index keys you want to see at the positions you specify. For example, SET-ORDER(1) positions the first sort or index key found in the dictionary for each set or index.

SET-ORDER(2) positions the second sort or index key, etc. If you use SET-ORDER without a subscript, it is equivalent to SET-ORDER(1).

When you use only the letter N, CA-IDMS/Schema Mapper places all sort or index keys in the set description beginning at the position indicated by SET-ORDER(N). The first sort or index key appears at that position followed by the remaining sort or index keys one per line.

You can choose to explicitly place one or more sort or index keys and have CA-IDMS/Schema Mapper implicitly place the remaining ones. Any sort or index keys that have not been positioned by an explicit subscript appear one per line, starting with the position indicated by SET-ORDER(N). See 2.9.6, “Sample XSETLINE Statements and Resulting Set Descriptions” on page 2-42.

When you do not use SET-ORDER(N) and there are more sort or index keys than you have explicitly positioned, the additional sort or index keys do not appear in the set description. An I011 message appears in the Audit Report to let you know this has happened. For example, if one of your sets or indexes has five sort or index keys, but you only use SET-ORDER(1), SET-ORDER(2), and SET-ORDER(3) in your XSETLINE statements, then the fourth and fifth sort or index keys do not appear in the set description.

Also, SET-ORDER(N) affects only sort or index keys that follow the highest explicit position. If any lower subscripts are omitted from the XSETLINE statements, the corresponding sort or index keys do not appear in the Cross-Reference Report. For example, if a set has five sort keys and the XSETLINE statements use SET-ORDER(1), SET-ORDER(3), and SET-ORDER(N), then SET-ORDER(N) positions sort keys 4 and 5 only. Sort key 2 does not appear in the Cross-Reference Report. Any sort or index key that does not appear in the Cross-Reference Report, for any reason, is listed in the Audit Report with an I011 message.

CA-IDMS/Schema Mapper makes all set descriptions the same size (remember, XSETLINE is a global parameter). Therefore, when only one of your sets or indexes contains multiple sort or index keys and you want all the keys for that set or index to appear in the set description, then every set description in the Cross-Reference Report contains space for multiple sort or index keys. This is true even if a particular set does not have a set order of sorted.

By default, CA-IDMS/Schema Mapper uses SET-ORDER(N). Therefore, all default set descriptions contain space for all sort or index keys.

2.9.6 Sample XSETLINE Statements and Resulting Set Descriptions

To Create a Three-Line Set Description:

To create a three-line set description containing the set name in line 1, the set order in line 2, and the pointers in line 3, use these parameter statements:

```
XSETLINE SET-NAME
XSETLINE SET-ORDER
XSETLINE POINTERS
```

A resulting set description looks like this:

```
5 DEPT-TEACHER      DEPT      39,  48 / AA      TEACHER      94,  48 / AA
                                     DEPT-TEACHER
                                     LAST
                                     NPO
```

To Create Blank Fields or Blank Lines in a Set Description:

You can create blank fields or blank lines in the set description to make room in the Cross-Reference Report for handwritten information. A set description with a blank five-character field in the first line and a blank second line is specified like this:

```
XSETLINE SET-NAME,BLANK 5,SET-ORDER
XSETLINE
```

Here are two resulting set descriptions:

```
5 DEPT-TEACHER      DEPT      39,  48 / AA      TEACHER      94,  48 / AA
                                     DEPT-TEACHER      LAST
6 IX-DEPT-NAME      SYSTEM (SR7)      DEPT      39,  48 / AA
                                     IX-DEPT-NAME      ASC DEPT-NAME DF
```

To Combine Explicit and Implicit Placement of Sort Keys:

A set description with the first sort key on the second line and additional sort keys on the last lines is specified like this:

```
XSETLINE SET-NAME
XSETLINE SET-ORDER(1)
XSETLINE POINTERS,MEM-OPT,SET-MODE
XSETLINE SET-ORDER(N)
```

Resulting set descriptions:

5	DEPT-TEACHER	DEPT	39,	48 / AA	TEACHER DEPT-TEACHER LAST NPO OM	94,	48 / AA
6	IX-DEPT-NAME	SYSTEM (SR7)			DEPT IX-DEPT-NAME ASC DEPT-NAME DF N MA MODE = INDEX	39,	48 / AA
7	IX-STUD-LNAME	SYSTEM (SR7)			STUDENT IX-STUD-LNAME DESC DBKEY DL N MA MODE = INDEX	94,	12 / AA

Use of SET-ORDER(N) always reserves a position in the set description, even when there are not enough sort or index keys to actually need it.

Field Names for the DSETLINE and XSETLINE Statements:

Field Name	Size (# of chars)	Descriptions
SET-NAME	16	The name of the set.
SET-ORDER	6	The order in which new records are positioned in the set: FIRST, LAST, NEXT, PRIOR, or sorted.
SET-MODE	13	The mode of the set. For regular sets the mode field is left blank. For indexed sets and integrated indexes, the mode field contains MODE = INDEX.
POINTERS	3	The set linkage options: N (next pointers), NP (next and prior pointers), NO (next and owner pointers), or NPO (next, prior, and owner pointers).
MEM-OPT (Membership Options)	2	The disconnect and connect options: MA (mandatory automatic), MM (mandatory manual), OA (optional automatic), or OM (optional manual).
BLANK nn	nn	Name used to specify a blank field.
UN-MEM-OCC-CNT	14	The count of member record occurrences of a record type within the set. The meaning differs slightly for multi-member sets.
UN-MEM-OCC-PCT	3	The percent of member record occurrences of a record type within the set. This is always 100% for single-member sets.

Field Name	Size (# of chars)	Descriptions
UN-SET-OCC	14	The count of total set occurrences for the set.
UN-SOWM-CNT	14	The count of set occurrences with members for the set.
UN-SOWM-PCT	3	The percent of set occurrences with members for the set.
UN-LOSO-AVG	17	The average length of set occurrences for the set.
UN-LOSO-MAX	14	The maximum length of set occurrences for the set.
UN-LOSO-MIN	14	The minimum length of set occurrences for the set.
UN-PG-SPRD-AVG	17	The average cluster page spread for the set.
UN-PG-SPRD-MAX	14	The maximum cluster page spread for the set.
UN-MEM-OVRF-CNT	14	The count of members on overflow for the set.
UN-MEM-OVRF-PCT	3	The percent of members on overflow for the set.
UN-MEM-OVRF-AVG	17	The average number of members on overflow for the set.
UN-MEM-OVRF-MAX	14	The maximum number of members on overflow for the set.
UN-MEM-OVRF-MIN	14	The minimum number of members on overflow for the set.
UN-PG-CHGS-AVG	17	The average page changes for the set.
UN-PG-CHGS-MAX	14	The maximum page changes for the set.
UN-PG-CHGS-MIN	14	The minimum page changes for the set.
UN-NUM-LVL	14	The number of levels for the index.
UN-BLS-UTE-CNT	14	The count of bottom level statistics used table entries for the index.
UN-BLS-UTE-AVG	17	The average bottom level statistics used table entries for the index.
UN-BLS-UTE-MAX	14	The maximum bottom level statistics used table entries for the index.

Field Name	Size (# of chars)	Descriptions
UN-BLS-UTE-MIN	14	The minimum bottom level used table entries for the index.

Exhibit 2.8: Field Names for the DSETLINE and XSETLINE Statements

2.10 INCLUDE AREA Statement

INCLUDE AREA is an optional parameter statement that allows you to represent a specific area in a diagram instead of the entire schema or subschema. The diagram obtained when the INCLUDE AREA statement is used represents all records contained in the specified area. The diagram also includes all indexes to those records and all sets that have both owners and members in the area. If a set has an owner or member outside the specified area, the set is not included in the diagram. Instead, the set is listed in the Cross-Reference Report.

2.10.1 INCLUDE AREA Syntax

[INCLUDE AREA *area name*]

where:

area name indicates the area from the schema or subschema to be represented by the diagram.

Rules:

- You must enter a separate INCLUDE AREA statement for each area. There is no limit to the number of INCLUDE AREA statements; you can enter one for every area in the schema or subschema.
- INCLUDE AREA statements must follow PROCESS, OPTIONS, CHARDEF, DRECLINE, XRECLINE, DSETLINE, and XSETLINE statements and precede LOCATE statements.

Default: If you do not use the INCLUDE AREA statement, you receive a diagram that represents every area within the schema or subschema.

2.10.2 When to Use the INCLUDE AREA Statement

The INCLUDE AREA statement can be used for these purposes:

- To control and reduce the size of the diagram when a schema or subschema is very large
- To create diagrams that are application-specific.

INCLUDE AREA also provides users who have generalized subschemas with the ability to diagram an area that is directly associated with a particular application.

2.11 LOCATE Statement

LOCATE, the final optional tailoring statement, allows you to control all or part of the layout of the diagram by specifying the position of record blocks. When you LOCATE a record block, you place it in a location by specifying how it should physically relate to another record block in the data structure diagram: you define its position in relative terms. You can position an individual record block in any direction, in units of record blocks, relative to another record block in the diagram.

2.11.1 LOCATE Syntax

```
[LOCATE name1 [[ / < LEFT > number \ ] [ / < UP > number \ ] FROM name2 ]]
```

name1

identifies, by REC-NAME, the record block you want to position in the diagram with a LOCATE statement. You cannot position the same record block twice during the same run of CA-IDMS/Schema Mapper.

```
/ < LEFT > number \
\ RIGHT /
```

number

specifies how many record blocks to the left or right of record *name2*, the record block identified by *name1* is to be positioned. You can use this parameter alone or with an UP or DOWN parameter. The number must be an integer between 1 and 10,000, inclusive.

```
/ < UP > number \
\ DOWN /
```

number

specifies how many record blocks up or down from record *name2*, the record block identified by *name1* is to be positioned. You can enter this parameter alone or with a LEFT or RIGHT parameter. The number must be an integer between 1 and 10,000, inclusive.

FROM name2

name2

specifies which record block is to be used as a reference for positioning the record block identified by *name1*. *name2* must be the REC-NAME of a record block that was used as *name1* in a previous LOCATE statement.

2.11.2 When to Use the LOCATE Statement

Here are a few reasons why you might want to position record blocks manually using the LOCATE statement:

- To produce a data structure diagram with a layout that resembles a familiar hand-drawn diagram
- To position key record blocks in a group
- To position a new record added to your schema or subschema, overriding CA-IDMS/Schema Mapper's automatic positioning feature.

2.11.3 Using the LOCATE Statement Most Efficiently

The LOCATE statement resembles a command, and it makes the layout of the diagram adjustable. How you use the LOCATE statement depends on what you want your diagram to look like and how much experience you have using CA-IDMS/Schema Mapper. Here are a few guidelines for using the LOCATE statement:

- First-time users--consider omitting the LOCATE statement. The first time you use CA-IDMS/Schema Mapper, it is a good idea not to use the LOCATE statement and allow CA-IDMS/Schema Mapper to position all of the record blocks automatically. This way, you can get a feel for how CA-IDMS/Schema Mapper positions the record blocks in your specific diagram. You can then effectively evaluate whether or not you need to use LOCATE statements.
- Experienced users--write LOCATE statements from scratch or edit LOCATE statements in the Transfer File. For the experienced user who is not satisfied with the automatic layout produced by CA-IDMS/Schema Mapper, writing LOCATE statements from scratch when first running CA-IDMS/Schema Mapper for a schema/subschema can be a time-saver. In this way, you can control the layout of a new diagram or duplicate the layout of an existing hand-drawn diagram right from the start.

If you are mainly concerned with the locations of a few key record blocks or a few groups of record blocks, you can write LOCATE statements for them from scratch and allow CA-IDMS/Schema Mapper to LOCATE the remaining record blocks in the diagram. If you want CA-IDMS/Schema Mapper to place record blocks between those that you LOCATE manually, you can leave plenty of space around them so CA-IDMS/Schema Mapper has room to work.

For users who have an existing diagram in hand and simply want to add one or two new record blocks, it may be easiest to add LOCATE statements to the Transfer File. If you just want to move around a few record blocks, it will probably be easiest to edit LOCATE statements in the Transfer File.

When the layout changes desired are extensive, positioning record blocks can be an iterating process, at least until you have worked with CA-IDMS/Schema Mapper enough to be able to know how it handles your particular schema/subschema. The

ideal approach depends on your needs. You can either write or edit LOCATE statements in the Transfer File or write LOCATE statements from scratch.

2.11.4 Graphic Considerations

When positioning only a few record blocks manually, results can be unpredictable. When using LOCATE statements, it is easiest to control the layout of the data structure diagram if you use CA-IDMS/Schema Mapper in a fully manual mode, writing LOCATE statements for all of the record blocks in the diagram.

When you use LOCATE to position only a few key record blocks, the interaction between your LOCATE statements and CA-IDMS/Schema Mapper's automatic placement of the remaining record blocks can create an unexpected layout. CA-IDMS/Schema Mapper maintains the relative positions of record blocks you LOCATE, but it also inserts record blocks into empty space created by your LOCATE statements (depending on the size of the space created). CA-IDMS/Schema Mapper does this regardless of whether the OPTIONS COMPRESS parameter is on or off (see following discussion on use of COMPRESS parameter).

If you use LOCATE statements and do not manually LOCATE every record block in the diagram, you probably need to execute CA-IDMS/Schema Mapper several times to get the desired layout.

Using the OPTIONS COMPRESS parameter while working on the layout of a diagram. If you will be executing CA-IDMS/Schema Mapper several times to obtain the desired layout of a diagram, you may want to turn off the OPTIONS COMPRESS parameter until the diagram nears completion.

When the COMPRESS parameter is off, CA-IDMS/Schema Mapper does not compress unused space from the diagram. CA-IDMS/Schema Mapper continues to insert record blocks into any empty space that it can.

When CA-IDMS/Schema Mapper creates the Transfer File and the COMPRESS parameter is on, CA-IDMS/Schema Mapper may rewrite some of the LOCATE statements you entered, changing the displacement values (the number of record blocks UP/DOWN/LEFT/RIGHT). CA-IDMS/Schema Mapper always maintains the relative positions of record blocks you LOCATE, but, with COMPRESS on, it may change the distances between some of the record blocks.

With the COMPRESS parameter off, CA-IDMS/Schema Mapper does not rewrite any of the LOCATE statements you entered as input. In other words, your LOCATE statements always appear unchanged in the Transfer File when you execute CA-IDMS/Schema Mapper with the COMPRESS parameter off. This can be an aid when you are working on the layout of the record blocks in a diagram.

Once you have the layout of the diagram near completion, you can turn on the COMPRESS parameter. CA-IDMS/Schema Mapper may rewrite any number of your LOCATE statements when it creates the Transfer File. CA-IDMS/Schema Mapper

compresses any unused space after all record blocks have been LOCATED in the diagram.

The size of an indexed record is larger than a record without indexes. An indexed record takes up more space on the diagram than a record without indexes. Because an index is represented by a diagonal line that projects from a corner of a record block, the record block is farther away from another record block allowing room for the indexes to be displayed.

Chapter 3. System Output

3.1 Data Structure Diagram	3-4
3.1.1 Basic Components of the Data Structure Diagram	3-4
3.1.2 Wallpapering: Technique Used to Assemble a Data Structure Diagram	3-6
3.1.3 How Large Will Your Diagrams Be?	3-9
3.2 Transfer File	3-10
3.2.1 Transfer File Statements	3-10
3.3 Cross-Reference Report	3-12
3.3.1 Cross-Reference Report Field Descriptions	3-12
3.4 Audit Report	3-15

This chapter gives an overview and describes the components of the output created by CA-IDMS/Schema Mapper. CA-IDMS/Schema Mapper produces four outputs:

- The data structure diagram
- The Transfer File
- The Cross-Reference Report
- The Audit Report.

3.1 Data Structure Diagram

The data structure diagram provides you with a graphic representation of your database's schema and/or subschemas. How your CA-IDMS/Schema Mapper data structure diagram actually looks depends on your schema or subschema and on the parameters you use to produce the diagram. The data structure diagram contains many more variables than the other outputs, giving you extensive control over the layout and the format. Exhibit 3.1 shows a simple data structure diagram created with automatic layout of the record blocks and sets and gives a brief description of the basic components.

3.1.1 Basic Components of the Data Structure Diagram

The main component of the CA-IDMS/Schema Mapper data structure diagram is the record block. Each rectangular-shaped graphic represents a CA-IDMS record and contains CA-IDMS record fields. The upper left corner of the diagram is the origin (0,0). The coordinate position of each record block, relative to this origin, appears in the Cross-Reference Report.

1. REC-NAME (record name) field identifies the name of the record type.
2. The LOC-MODE (location mode) field identifies how records are physically located in their areas: CALC, VIA, or DIRECT.
3. The LENGTH-MODE field identifies the record modes: F (fixed), V (variable), FC (fixed compressed), or VC (variable compressed). The value for this field is not stored in the dictionary in any literal sense. CA-IDMS/Schema Mapper determines the value as follows:
 - A DEPENDS ON data element in the record indicates a variable length mode. Otherwise, the record is assumed to have a fixed length mode.
 - The appearance of IDMSCOMP or IDMSDCOM in the list of procedures, which are to be invoked when the record is accessed, indicates a compressed length mode. Otherwise, the record is assumed to have an uncompressed length mode.
4. The LENGTH (record size) field identifies the actual data length of the record (if fixed length), or the maximum data length of the record (if variable length), in bytes. If the value obtained from the dictionary is less than 1, CA-IDMS/Schema Mapper uses a value of 1 in the record block. This happens, for example, when a schema record is defined without any elements and is consistent with the value reported by IDMSRPTS for data length.
5. The AREA field identifies the name of the area in which the record type is located.
6. The DUPS-OPT (duplicates option) field identifies the disposition of records with duplicate control keys: DN (duplicates not allowed), DF (duplicates first), or DL (duplicates last). This field is blank for non-CALC records.
7. The LOC-CTRL (location control) field identifies the CALC key name or VIA set name. This is always DIRECT-DBK for DIRECT records.

8. The REC-ID (record identification) field gives the unique identification number of the record type.
9. Groups of record blocks are connected by set connection lines, which show set relationships. Sets are identified by numbers in the CA-IDMS/Schema Mapper data structure diagram and are cross-referenced in the Cross-Reference Report. A unique number from 1 to n is assigned to each set, and the unique number appears with each set connection in the diagram as described below:
 - When the set connection is a straight line, the number appears near the owner
 - When the set connection has a set turn in it, the number appears near the owner and near the member
 - For multi-member sets, the number appears near the owner and near each member of the set.

Record blocks that represent members of sets have arrows pointing toward them. Record blocks that represent owners of sets have arrows pointing away from them.

10. The diagonal line that appears in the diagram for indexes contains the letters II when the line represents an integrated index.

3.1 Data Structure Diagram

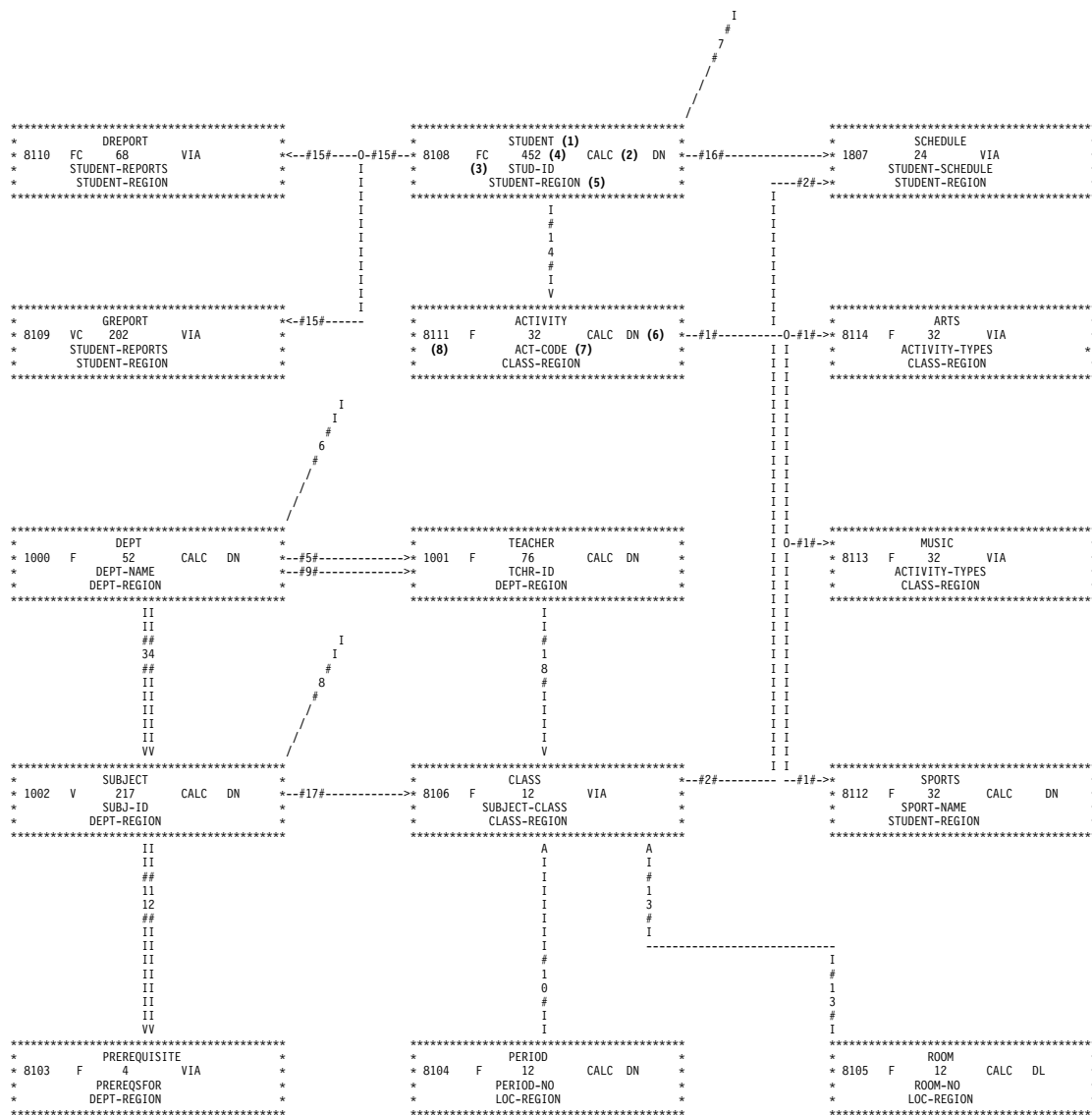


Exhibit 3.1: Sample of a Simple Data Structure Diagram

3.1.2 Wallpapering: Technique Used to Assemble a Data Structure Diagram

Data structure diagrams are typically large, so the printed output spans multiple pages. CA-IDMS/Schema Mapper automatically divides the overall diagram into pieces that are the size of your printer's paper, with each piece being the size of a page. You assemble the pages in columns down the length of the diagram and rows across the width of the diagram. (See Exhibit 3.2)

The diagram is printed sequentially, one column of pages after another, beginning with the column that runs down the left side of the diagram (column A). CA-IDMS/Schema

Mapper automatically generates column headers at the top of each column to help you find the beginning of the printout for each column. Your printer will put a blank page after the last page of each column. This blank page helps eliminate wallpapering problems if your diagram does not begin printing right at the top of a page.

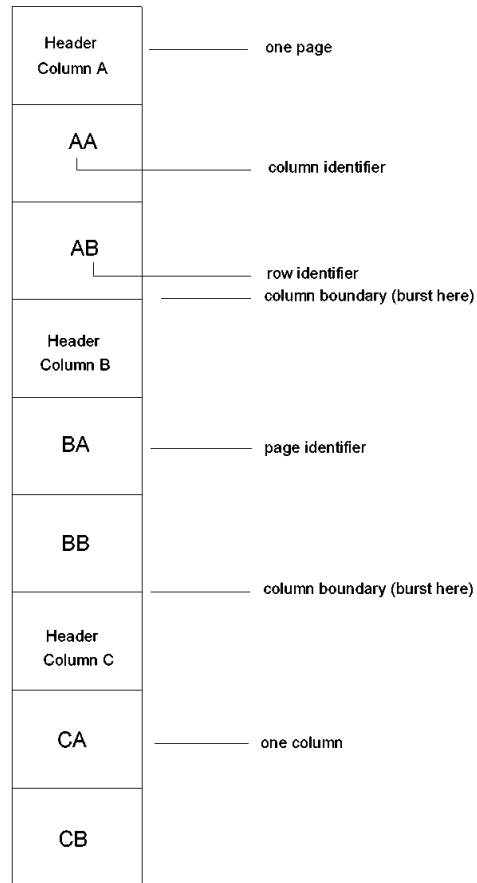
A unique two-character (alphabetic) page identifier is also automatically printed, in the upper right corner of each page. The first character identifies the page's column in the overall diagram, and the second character identifies its row in the overall diagram. The first page of the diagram (the top left corner) is in the first column, and the first row and is identified by the characters AA.

To assemble the data structure diagram:

- Manually burst the printout at the column boundaries (at the top of the column headers).
- Fold each column so the column header is on top.
- Using a paper cutter or a pair of scissors, trim off the left edge of the entire diagram printout, using the asterisks (*) as a guide. Be careful not to cut into the asterisks, or you may cut off part of the diagram.
- Align each column to form the horizontal rows, and tape the columns together.

3.1 Data Structure Diagram

Print out of Data Structure
Diagram Before Bursting



The Data Structure Diagram
After Bursting and Wallpapering

Header Column A	Header Column B	Header Column C
AA	BA	CA
AB	BB	CB

Exhibit 3.2: Wallpapering the CA-IDMS/Schema Mapper Data Structure Diagram

3.1.3 How Large Will Your Diagrams Be?

The size of the data structure diagram primarily depends on the size of the schema or subschema being diagrammed. Other factors that can affect the size of a diagram are:

- The configuration of the schema or subschema
- The size of each record block and set description, as determined by the number, length, and organization of fields
- The border space around record blocks.

The size of CA-IDMS/Schema Mapper's CA-IDMS IDMSNWKA diagram may give you an idea of a diagram's size. IDMSNWKA (the subschema for the CA-IDMS Network A database) is eight pages by eight pages (nine feet by seven feet, using 11" by 14" perforated paper). It has 151 records and 227 sets.

If your diagrams of schemas seem too large, you can run CA-IDMS/Schema Mapper for areas or subschemas rather than schemas. Another option, which has good results in reducing the size of diagrams, is to photo reduce the diagrams.

The maximum size that an assembled CA-IDMS/Schema Mapper data structure diagram can be is 26 pages wide by 26 pages long. You probably won't have any diagrams even close to this size.

Assuming a DPAGEWID value of 132 and a DPAGELEN value of 66, the maximum dimensions allow 3,354 characters across (26 pages by 129 characters) and 1,716 characters down (26 pages by 66 characters) from the initial reference point.

The right three character positions of each line are reserved for page identifiers, such as AA, other information, such as the date and schema/subschema name, and a blank character space to separate this information from the diagram.

3.2 Transfer File

The Transfer File is an output file that contains parameter statements. This file can make it easier and less time-consuming for you to recreate or modify a previously-created CA-IDMS/Schema Mapper data structure diagram. By using the parameter statements in the Transfer File as input, you can recreate a diagram without re-keying the parameter statements. You can also edit the statements in the Transfer File to tailor a diagram.

The Transfer File contains a copy of the parameter statements used as input. The Transfer File always contains a PROCESS statement, and if they were specified, OPTIONS, CHARDEF, DRECLINE, XRECLINE, DSETLINE, XSETLINE, INCLUDE, and LOCATE statements. PROCESS, OPTIONS, CHARDEF, DRECLINE, XRECLINE, DSETLINE, XSETLINE, and INCLUDE statements are transferred (or copied) to the Transfer File without being modified by CA-IDMS/Schema Mapper. If the input does not contain LOCATE statements, CA-IDMS/Schema Mapper writes a LOCATE statement for every record block in the diagram.

3.2.1 Transfer File Statements

Following is a description of the Transfer File statements shown in Exhibit 3.3.

Statements always contained in the Transfer File

1. A PROCESS statement, identical to the one in the parameters used as input, identifying the schema or subschema to be represented by the diagram.
2. LOCATE statements, copied or written by CA-IDMS/Schema Mapper, identifying the positions of all record blocks in the diagram.

These statements are contained in the Transfer File, if specified in the parameters used as input

3. An OPTIONS statement that specifies layout options, including page size and the compression of unused space, and whether or not Index Set records are included in the diagram.
4. A CHARDEF statement that primarily controls printer-dependent options.
5. A DRECLINE statement that specifies what information is included in the record blocks in the data structure diagram.
6. An XRECLINE statement that specifies what information is included in the record block description in the Cross-Reference Report.
7. A DSETLINE statement that specifies what information is included in the set and index descriptions in the data structure diagram.
8. An XSETLINE statement that specifies what information is included in the set and index descriptions in the Cross-Reference Report.
9. An INCLUDE statement that specifies the area to be represented by the diagram.


```

1 ► PROCESS=SUBSCHEMA,SCHEMA=DSRR1016,VERSION=1,
    SUBSCHEMA=DBRR1016, DICTNAME=TKIT
3 ► OPTIONS,COMPRESS=ON
4 ► CHARDEF,RHORCHAR=*
5 ► DRECLINE BLANK 3,REC-NAME
    DRECLINE REC-ID,LENGTH-MODE,LENGTH,BLANK 12,LOC-MODE,DUPS-OPT
    DRECLINE BLANK 3,LOC-CTRL
    DRECLINE BLANK 3,AREA
    DRECLINE REC-DESC
6 ► XRECLINE REC-NAME
    XRECLINE AREA,LOC-MODE
    XRECLINE LENGTH,REC-DESC
7 ► DSETLINE SET-NAME,BLANK 5,SET-ORDER
    DSETLINE
8 ► XSETLINE SET-NAME
    XSETLINE POINTERS,BLANK 4,MEM-OPT
    XSETLINE
    XSETLINE SET-ORDER
9 ► INCLUDE AREA STUDENT-REGION
    INCLUDE AREA CLASS-REGION
    INCLUDE AREA DEPT-REGION
    INCLUDE AREA LOC-REGION
$$$ NO LOCATE STATEMENTS WERE PROVIDED BY THE USER $$$
$$$ LOCATE STATEMENTS GENERATED BY CA-IDMS/Schema Mapper $$$
2 ► LOCATE STUDENT
    LOCATE ACTIVITY                DOWN      1  FROM STUDENT
    LOCATE ARTS                     RIGHT     1  DOWN      1  FROM ACTIVITY
    LOCATE MUSIC                    RIGHT     1  DOWN      1  FROM ACTIVITY
    LOCATE SPORTS                   RIGHT     1  DOWN      2  FROM ACTIVITY
    LOCATE DREPORT                  LEFT      1  FROM STUDENT
    LOCATE GREPORT                  LEFT      1  DOWN      1  FROM STUDENT
    LOCATE SCHEDULE                  RIGHT     1  FROM STUDENT
    LOCATE DEPT                     LEFT      1  DOWN      2  FROM STUDENT
    LOCATE SUBJECT                  DOWN      1  FROM DEPT
    LOCATE PREREQ                   DOWN      1  FROM SUBJECT
    LOCATE CLASS                    RIGHT     1  FROM SUBJECT
    LOCATE TEACHER                  RIGHT     1  FROM DEPT
    LOCATE PERIOD                   DOWN      1  FROM CLASS
    LOCATE ROOM                     RIGHT     1  DOWN      1  FROM CLASS
    LOCATE CON-TROL                 LEFT      2  FROM STUDENT

```

Exhibit 3.3: Sample Statements Contained in the Transfer File

3.3 Cross-Reference Report

The Cross-Reference Report is a listing for quickly finding individual record blocks, sets, and indexes in a CA-IDMS/Schema Mapper data structure diagram. The Cross-Reference Report includes descriptions of records, sets, and indexes. You can control record block and set and index descriptions by using the optional XRECLINE and XSETLINE statements, respectively. The use of XRECLINE and XSETLINE can affect this output significantly.

The Cross-Reference Report references sets and indexes by their unique number assigned by CA-IDMS/Schema Mapper. The numbers appear in the CA-IDMS/Schema Mapper data structure diagram within the set connection lines, close to the owners, members, and/or indexes.

The Cross-Reference Report references the locations of record blocks in the data structure diagram by both page identifier and coordinate position. The page identifier is a two-character (alphabetic) identifier that shows on which page (the column and row) the upper left corner of a record block can be found in the data structure diagram. A record block that is referenced in the Cross-Reference Report by the page identifier AB is in the first (or A) column that runs down the length of the diagram and the second (or B) row that runs across the width of the diagram.

The coordinate position is used to find the exact location of a record block in the CA-IDMS/Schema Mapper data structure diagram. A coordinate position is a combination of two numbers that refer to the position of the upper left corner of each record block in the diagram. The positions are numbered in units of character spaces, with the upper left corner of the diagram being the origin (0,0). The first number tells how many character spaces the record is from the left side of the diagram. The second number tells how many character spaces the record block is from the top of the diagram.

3.3.1 Cross-Reference Report Field Descriptions

Field descriptions for the Cross-Reference Report (Exhibit 3.4) are described below:

RECORD NAME field contains the REC-NAME of each record block in the CA-IDMS/Schema Mapper data structure diagram.

LOCATION field contains the location of each record block in the diagram, given by coordinate position.

PAGE field contains the location of each record block in the diagram, given by page identifier.

SET NUMBER field contains the unique number assigned to each set and index in the diagram.

SET OR INDEX NAME field contains the SET-NAME of each set and index in the diagram.

OWNER field contains the OWNER record names for each set. Integrated indexes are labeled SYSTEM(SR7).

LOCATION field contains the locations of the OWNER records, given by coordinate position.

PAGE field contains the location of OWNER records, given by page identifier.

MEMBER field contains the record names for each set and the following: the set name, set linkage options (N for next pointers, NP for next and prior pointers, NO for next and owner pointer, or NPO for next, prior, and owner pointers), disconnect and connect options (MA for mandatory automatic, MM for mandatory manual, OA for optional automatic, or OM for optional manual), and the order in which new records are positioned in the set (FIRST, LAST NEXT, PRIOR, or sorted).

LOCATION field contains the locations of the MEMBER records, given by coordinate position.

PAGE field contains the locations of the MEMBER records, given by page identifier.

3-14 CA-IDMS Schema Mapper User Guide

Exhibit 3.4: Sample Cross-Reference Report

3.4 Audit Report

The Audit Report contains a summary of all processing performed and a listing of all informative, warning, and error messages generated during program execution. See Chapter 6, “Messages” on page 6-1 for a listing of all possible messages and suggested actions to be taken.

The Audit Report (Exhibit 3.5) contains:

1. Processing start date and time.
2. Informative messages that report processing starts and stops (and actions taken), and also provide a list of the parameters used.
3. Warning messages that report any conflicting parameter data encountered, and CA-IDMS/Schema Mapper's corrective actions.
4. Processing stop date and time.

CA-IDMS/Schema Mapper		nn.nn	mm/dd/yy	COMPUTER ASSOCIATES	PAGE 2
AUDIT REPORT					
1 ▶	I001	Schema Mapper STARTED mm/dd/yy hh:mm:ss.			
2 ▶	I002	PARAMETER INPUT		PROCESS=SUBSCHEMA,SCHEMA=DSRR1016,VERSION=1,	
	I002	PARAMETER INPUT		SUBSCHEMA=DBRR1016,DICTIONAME=TKIT	
	I002	PARAMETER INPUT		OPTIONS,COMPRESS=ON	
	I010	Schema Mapper IS RUNNING UNDER IDMS VERSION nn.nn			
	I007	THIS SCHEMA WAS COMPILED USING IDMS nn.nn <????>.			
	I009	USING SCHEMA VERSION 1.			
	I008	THIS SCHEMA WAS COMPILED ON mm/dd/yy.			
	I002	PARAMETER INPUT		CHARDEF,RHORCHAR==,RVERCHAR=;	
	I002	PARAMETER INPUT		DRECLINE REC-DESC	
3 ▶	W004	MINIMUM FRAGMENT SPECIFIED FOR NON-COMPRESSED FIXED RECORD		MUSIC.	
	W004	MINIMUM FRAGMENT SPECIFIED FOR NON-COMPRESSED FIXED RECORD		ARTS.	
	W004	MINIMUM FRAGMENT SPECIFIED FOR NON-COMPRESSED FIXED RECORD		SPORTS.	
	I002	PARAMETER INPUT		DRECLINE AREA,REC-ID,LOC-MODE	
	I002	PARAMETER INPUT		DRECLINE LOC-CTRL	
	I002	PARAMETER INPUT		XSETLINE SET-NAME	
	I002	PARAMETER INPUT		XSETLINE POINTERS,MEM-OPT	
	I002	PARAMETER INPUT		INCLUDE AREA STUDENT-REGION	
	I002	PARAMETER INPUT		INCLUDE AREA CLASS-REGION	
	I002	PARAMETER INPUT		LOCATE STUDENT	
	I002	PARAMETER INPUT		LOCATE CLASS DOWN 2 FROM STUDENT	
	I006	GENERATED STATEMENT	LOCATE ACTIVITY	DOWN	1 FROM STUDENT
	I006	GENERATED STATEMENT	LOCATE ARTS	RIGHT	1 FROM ACTIVITY
	I006	GENERATED STATEMENT	LOCATE MUSIC	RIGHT	1 DOWN 1 FROM ACTIVITY
	I006	GENERATED STATEMENT	LOCATE SPORTS	RIGHT	1 DOWN 2 FROM ACTIVITY
	I006	GENERATED STATEMENT	LOCATE DREPORT	LEFT	1 FROM STUDENT
	I006	GENERATED STATEMENT	LOCATE GREPORT	LEFT	1 DOWN 1 FROM STUDENT
	I006	GENERATED STATEMENT	LOCATE SCHEDULE	RIGHT	1 DOWN FROM STUDENT
	I011	SKIPPING KEY NUMBER 1 IN CLASS-SCHEDULE SET...			
	I012	...KEY NAME IS ASC ROSTER-NO DL OF MEMBER SCHEDULE.			
	I011	SKIPPING KEY NUMBER 1 IN IX-STUD-LNAME INDEX...			
	I012	...KEY NAME IS DESC DBKEY DL OF MEMBER STUDENT.			
	I011	SKIPPING KEY NUMBER 1 IN STUDENT-SCHEDULE SET...			
	I012	...KEY NAME IS ASC DBKEY DN OF MEMBER SCHEDULE.			
4 ▶	I003	Schema Mapper ENDED mm/dd/yy hh:mm:ss.			

Exhibit 3.5: Sample Audit Report

Chapter 4. Examples

4.1	Example 1	4-4
4.2	Example 2	4-7
4.3	Example 3	4-9
4.4	Example 4	4-13
4.5	Example 5	4-15

This chapter gives examples of output created with CA-IDMS/Schema Mapper. It gives examples of how optional parameters affect the diagrams and reports. Each example includes the parameter statements entered, an explanation of the task performed, and an illustration of the resulting output. Exhibit 4.1 lists the examples in this chapter.

Example	Parameter Statements	Description
1	PROCESS only	To produce a data structure diagram and Cross-Reference Report by default.
2	CHARDEF and INCLUDE	To change the print characters used to draw the diagram and to represent only specified areas from the schema or subschema.
3	DRECLINE and XSETLINE	To modify the record blocks in the diagram and set descriptions in the Cross-Reference Report.
4	LOCATE	To manually position record blocks in the diagram.
5	PROCESS only	To produce output with multiple CALC and sort keys.

Exhibit 4.1: List of Examples

4.1 Example 1

The data structure diagram and Cross-Reference Report in Exhibits 4.2 thru 4.5 were produced the first time the user ran CA-IDMS/Schema Mapper. The user wanted to see the automatic default output and then determine whether to tailor it.

Parameters Entered--The user entered a single PROCESS statement to request that CA-IDMS/Schema Mapper draw the data structure diagram automatically and according to default values.

Exhibit 4.2 lists the parameter statement used and shows the resulting Transfer File. You can use the Transfer File as input to create the identical diagram at a later time, and you can edit it to tailor the output.

Note: The output shown in Examples 2 through 4 are tailored variations of the output in this example.

```
PROCESS=SUBSCHEMA, SCHEMA=TSDSCHM1, VERSION=1, SUBSCHEMA=TSDSUBS1,
DICTNAME=SMAP
```

Exhibit 4.2: Example 1--Parameters Entered

```
PROCESS=SUBSCHEMA, SCHEMA=TSDSCHM1, VERSION=1, SUBSCHEMA=TSDSUBS1,
DICTNAME=SMAP
$$$ NO LOCATE STATEMENTS WERE PROVIDED BY THE USER          $$$
$$$ LOCATE STATEMENTS GENERATED BY Schema Mapper            $$$
LOCATE STUDENT
LOCATE ACTIVITY                DOWN      1    FROM STUDENT
LOCATE ARTS                    RIGHT     1    FROM ACTIVITY
LOCATE MUSIC                   RIGHT     1    DOWN    1    FROM ACTIVITY
LOCATE SPORTS                  RIGHT     1    DOWN    2    FROM ACTIVITY
LOCATE DREPORT                 LEFT      1    FROM STUDENT
LOCATE GREPORT                 LEFT      1    DOWN    1    FROM STUDENT
LOCATE SCHEDULE                RIGHT     1    FROM STUDENT
LOCATE DEPT                    LEFT      1    DOWN    2    FROM STUDENT
LOCATE SUBJECT                 DOWN      1    FROM DEPT
LOCATE PREREQ                  DOWN      1    FROM SUBJECT
LOCATE CLASS                   RIGHT     1    FROM SUBJECT
LOCATE TEACHER                 RIGHT     1    FROM DEPT
LOCATE PERIOD                  DOWN      1    FROM CLASS
LOCATE ROOM                    RIGHT     1    DOWN    1    FROM CLASS
LOCATE CON-TROL                LEFT      2    DOWN    1    FROM STUDENT
```

Exhibit 4.3: Example 1--Content of the Resulting Transfer File

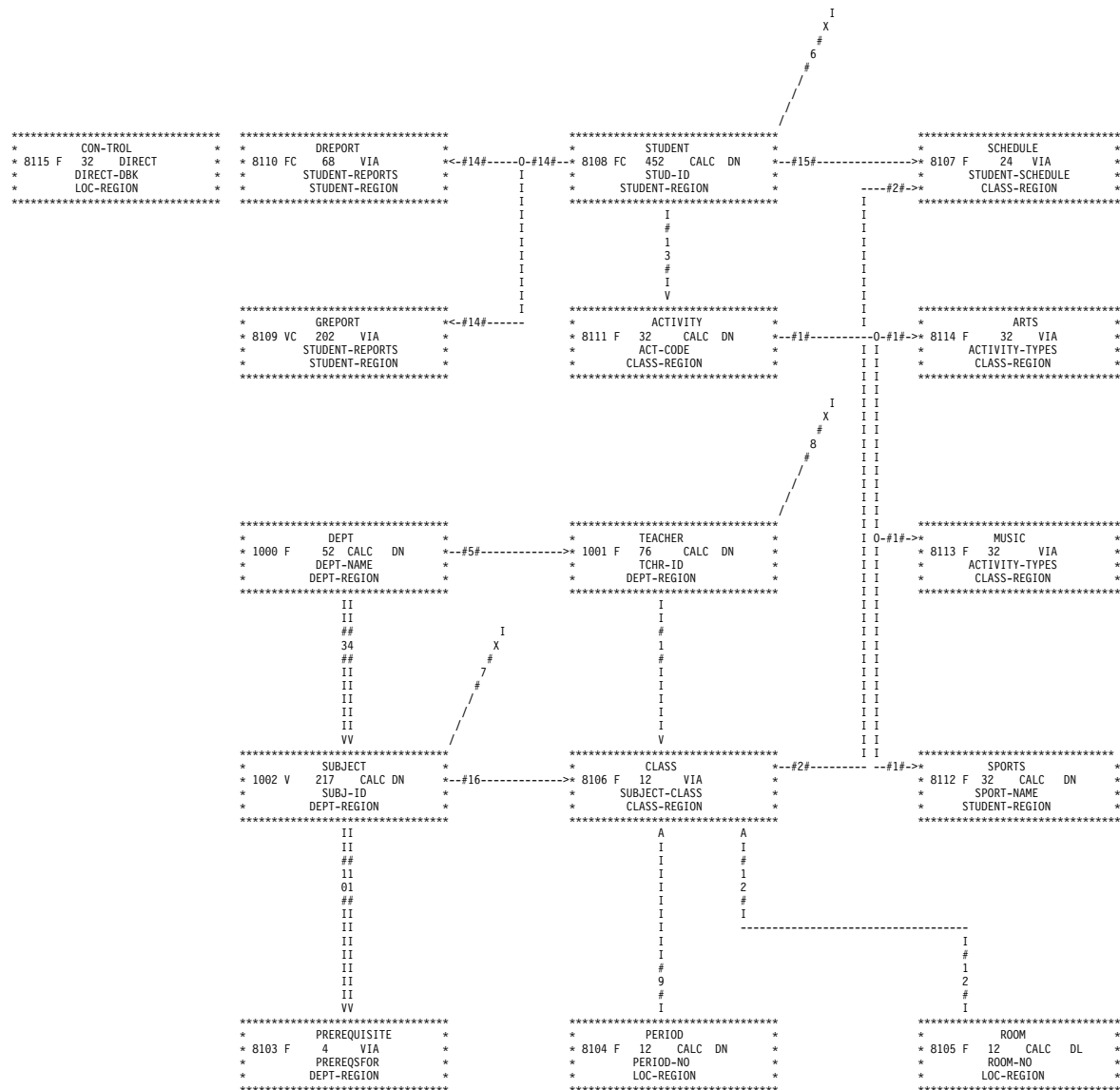


Exhibit 4.4: Default CA-IDMS/Schema Mapper Data Structure Diagram

4.1 Example 1

CA-IDMS/Schema Mapper: mm/dd/yy hh:mm:ss SUBSCHEMA DBRR1016, SCHEMA DSRR1016, VERSION 1, DICTNAME=TKIT, DBNAME=											
RECORD CROSS-REFERENCE LISTING											
RECORD NAME		LOCATION		/		PAGE					
-----		-----		-----		-----					
ACTIVITY		94,		27		/		AA			
ARTS		150,		27		/		BA			
CLASS		94,		74		/		AB			
CON-TROL		1,		12		/		AA			
DEPT		39,		48		/		AA			
DREPORT		39,		12		/		AA			
GREPORT		39,		27		/		AA			
MUSIC		150,		48		/		BA			
PERIOD		94,		95		/		AB			
PREREQ		39,		95		/		AB			
ROOM		150,		95		/		BB			
SCHEDULE		150,		12		/		BA			
SPORTS		150,		74		/		BB			
STUDENT		94,		12		/		AA			
SUBJECT		39,		74		/		AB			
TEACHER		94,		48		/		AA			

CA-IDMS/Schema Mapper: mm/dd/yy hh:mm:ss SUBSCHEMA DBRR1016, SCHEMA DSRR1016, VERSION 1, DICTNAME=TKIT, DBNAME=											
SET/INDEX CROSS-REFERENCE LISTING											
SET	SET OR INDEX NAME	OWNER	LOCATION		/	PAGE	MEMBER	LOCATION	/	PAGE	
-----			-----	-----			-----	-----	-----	-----	-----
1	ACTIVITY-TYPES	ACTIVITY	94,	27	/	AA	ARTS	150,	27	/	BA
							ACTIVITY-TYPES				
							NP OA				
							LAST				
							MUSIC	150,	48	/	BA
							ACTIVITY-TYPES				
							NP OA				
							LAST				
							SPORTS	150,	74	/	BB
							ACTIVITY-TYPES				
							NPO OM				
							LAST				
2	CLASS-SCHEDULE	CLASS	94,	74	/	AB	SCHEDULE	150,	12	/	BA
							CLASS-SCHEDULE				
							NPO MA				
	ASC ROSTER-NO DL										
3	DEPT-ACADEMIC	DEPT	39,	48,	/	AA	SUBJECT	39,	74	/	AB
							DEPT-ACADEMIC				
							NO MM				
	NEXT										
4	DEPT-GENERAL	DEPT	39,	48	/	AA	SUBJECT	39,	74	/	AB
							DEPT-GENERAL				
							NO MM				
	NEXT										
5	DEPT-TEACHER	DEPT	39,	48	/	AA	TEACHER	94,	48	/	AA
							DEPT-TEACHER				
							NPO OM				
	LAST										
6	IX-DEPT-NAME	SYSTEM (SR7)	39,	48	/	AA	DEPT	39,	48	/	AA
							IX-DEPT-NAME				
							N MA MODE = INDEX				
	ASC DEPT-NAME DF										

Exhibit 4.5: Cross-Reference Report with Default Values

4.2 Example 2

Example 2 shows how the user changed the print characters used to draw the data structure diagram. This example also illustrates the INCLUDE statement by representing only the areas that the user specified instead of representing the entire schema or subschema. (See Exhibit 4.7.)

Why the print characters were changed--The user knew that the default characters used to draw the left and right arrows in the diagram, the less-than (<) and greater-than (>) symbols, were not on the user's print chain. In addition, the user wanted to experiment with the characters used to draw record blocks, set connections, and indexes.

Why specific areas were selected--Instead of representing the entire schema or subschema by the diagram, the user needed only the areas specified to be represented in the diagram for a particular application. The INCLUDE statement produced a diagram that represents only the areas the user selected.

Parameters Entered--In this case, the user wrote PROCESS, CHARDEF, and INCLUDE statements from scratch to a Parameter Statements File (see Exhibit 4.1).

```
PROCESS=SUBSCHEMA, SCHEMA=DSRR1016, VERSION=1, SUBSCHEMA=DBRR1016,  
    DICTNAME=TKIT  
  
OPTIONS,COMPRESS=ON  
  
CHARDEF,RHORCHAR==,RVERCHAR=;,SLLCHAR=L,SLRCHAR=J,  
    SULCHAR=F,SURCHAR=7,LARROW=C,RARROW=D,ISLACHAR=X  
  
INCLUDE AREA STUDENT-REGION  
INCLUDE AREA CLASS-REGION
```

Exhibit 4.6: Example 2--Parameters Entered

4.2 Example 2

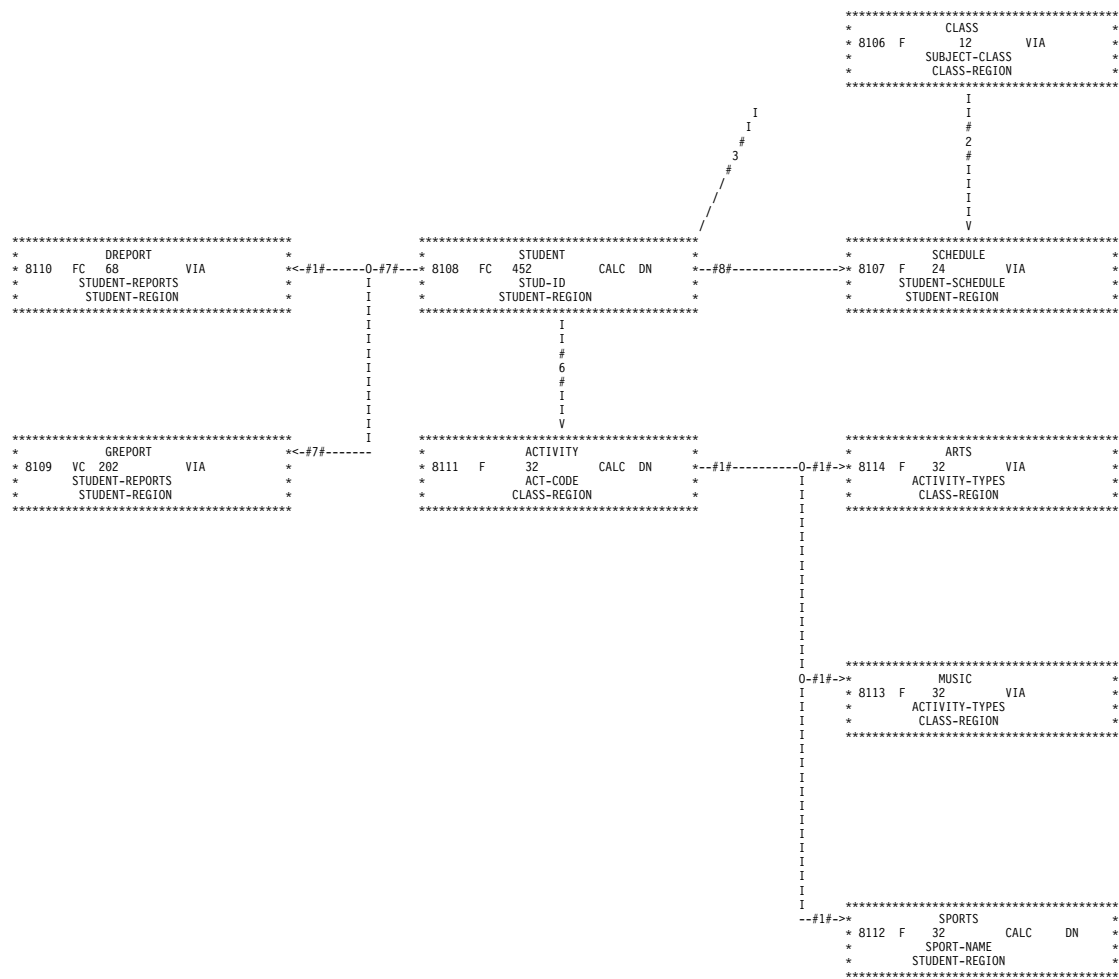


Exhibit 4.7: Tailored CA-IDMS/Schema Mapper Data Structure Diagram with User-Specified Characters and Areas

4.3 Example 3

Here, the user wanted to modify the format of record blocks in the CA-IDMS/Schema Mapper data structure diagram and the descriptions of sets in the Cross-Reference Report (see Exhibits 4.9 and 4.10).

Why record blocks and set descriptions were modified--The user's predefined record block formats and set descriptions differed slightly from CA-IDMS/Schema Mapper's default specifications. Also, the REC-NAME field does not meet the user's requirements, but the user wants to include a more meaningful record description in the record block. The user did this by including the REC-DESC field and choosing these fields for the record block:

- A blank field (for aesthetic reasons) between the LENGTH field and the LOC-MODE field in the second line of the record block
- A blank line for handwritten notes for the third line of the record block
- A REC-DESC field for descriptive names for the fourth line of the record block.

The user did not want the LOC-CTRL and AREA fields included in the record block, so the user did not enter them in a DRECLINE statement. (See Exhibit 4.8.)

Note: The user's modifications to the format of the record block also change the size and the shape of the record blocks in the diagram.

For set descriptions, the user wanted:

- A blank field between POINTERS and MEM-OPT
- A blank line for handwritten notes for the third line of the set description.

Also, the user did not want the SET-ORDER field included in the set description, so the user did not enter an XSETLINE statement for it. (See Exhibit 4.8.)

Parameters Entered--The user wrote DRECLINE and XSETLINE statements from scratch to a Parameters Statements File to globally modify the record block format and the set description (see Exhibit 4.8).

Note: When the user specified DRECLINE and XSETLINE parameters, all defaults for the record block and the set description were canceled. The user had to rebuild the entire record block (content) with DRECLINE and the entire set description with XSETLINE.

4.3 Example 3

```
PROCESS=SUBSCHEMA, SCHEMA=DSRR1016, SUBSCHEMA=DBRR1016, VERSION=1,  
DICTNAME=TKIT  
  
OPTIONS,COMPRESS=ON  
  
CHARDEF,RHORCHAR==,RVERCHAR=: ,SLLCHAR=L,SLRCHAR=J,  
SULCHAR=F,SURCHAR=7,LARROW=C,RARROW=D,ISLACHAR=X  
  
DRECLINE REC-NAME  
DRECLINE REC-ID,BLANK 5,LOC-MODE,DUPS-OPT  
DRECLINE LENGTH-MODE,LENGTH  
DRECLINE LABEL 'COUNT',UN-NUM-OCC  
  
XSETLINE SET-NAME  
XSETLINE POINTERS,BLANK 8,MEM-OPT  
XSETLINE
```

Exhibit 4.8: Example 3--Parameters Entered

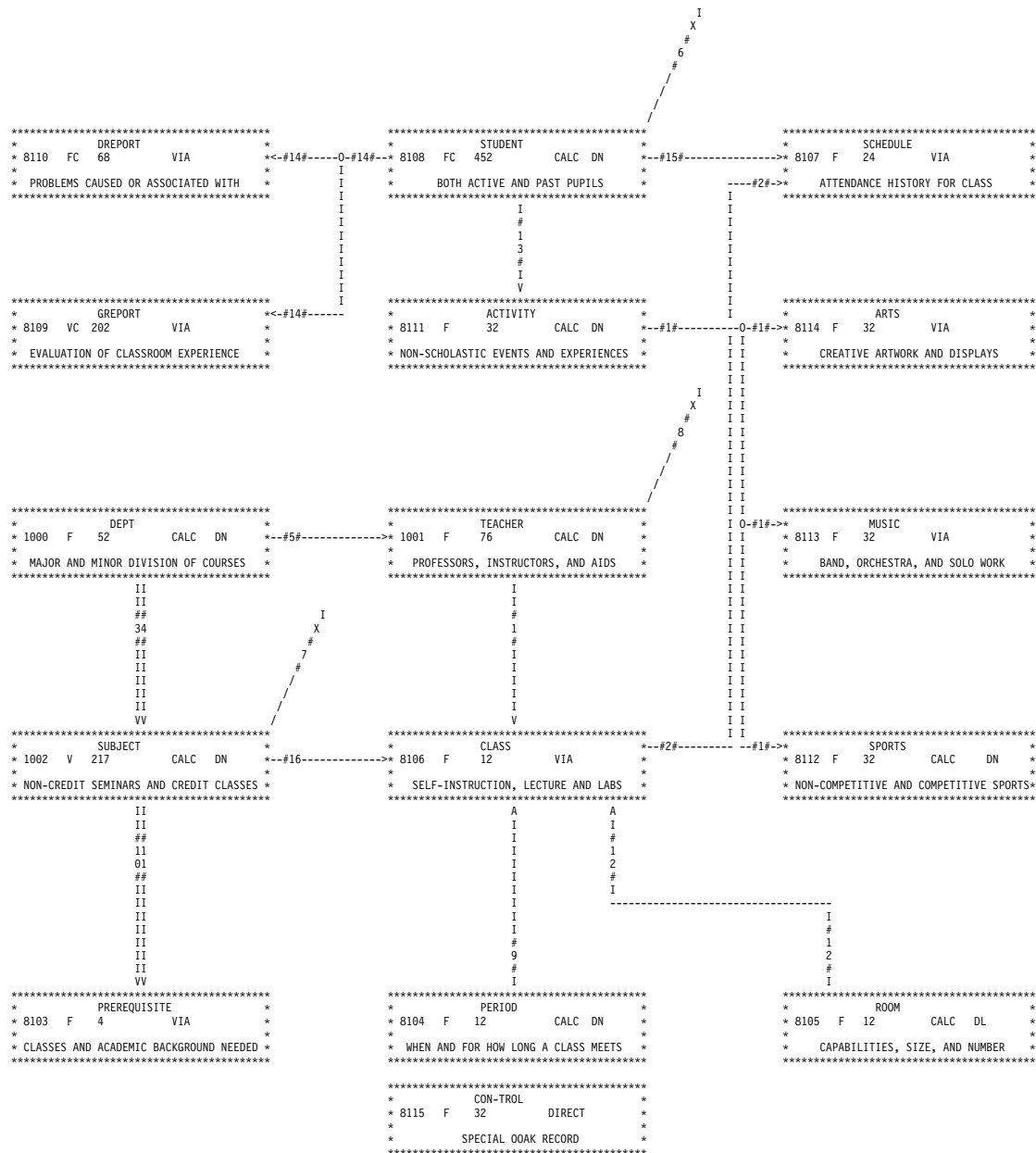


Exhibit 4.9: Tailored CA-IDMS/Schema Mapper Data Structure Diagram with Record Blocks Modified

4.3 Example 3

CA-IDMS/SCHEMA MAPPER: mm/dd/yy hh:mm:ss SUBSCHEMA DBRR1016, SCHEMA DSRR1016, VERSION 1, DICTNAME=TKIT, DBNAME=										
RECORD CROSS-REFERENCE LISTING										
RECORD NAME	LOCATION			/	PAGE					
-----	-----			-----	-----					
ACTIVITY	70,	27	/	AA						
ARTS	114,	27	/	BA						
CLASS	70,	74	/	AB						
CON-TROL	1,	12	/	AA						
DEPT	27,	48	/	AA						
DREPORT	27,	12	/	AA						
GREPORT	27,	27	/	AA						
MUSIC	114,	48	/	BA						
PERIOD	70,	95	/	AB						
PREREQ	27,	95	/	AB						
ROOM	114,	95	/	BB						
SCHEDULE	114,	12	/	BA						
SPORTS	114,	74	/	BB						
STUDENT	70,	12	/	AA						
SUBJECT	27,	74	/	AB						
TEACHER	70,	48	/	AA						
CA-IDMS/SCHEMA MAPPER: mm/dd/yy hh:mm:ss SUBSCHEMA DBRR1016, SCHEMA DSRR1016, VERSION 1, DICTNAME=TKIT, DBNAME=										
SET/INDEX CROSS-REFERENCE LISTING										
SET NUMBER	SET OR INDEX NAME	OWNER	LOCATION		/	PAGE	MEMBER	LOCATION	/	PAGE
-----	-----	-----	-----		-----	-----	-----	-----	-----	-----
1	ACTIVITY-TYPES	ACTIVITY	70,	27	/	AA	ARTS	114, 27	/	BA
							ACTIVITY-TYPES NPO OA			
							MUSIC	114, 48	/	BA
							ACTIVITY-TYPES NPO OA			
							SPORTS	114, 74	/	BB
							ACTIVITY-TYPES NPO OM			
2	CLASS-SCHEDULE	CLASS	70,	74	/	AB	SCHEDULE	114, 12	/	BA
							CLASS-SCHEDULE NPO MA			
3	DEPT-ACADEMIC	DEPT	27,	48,	/	AA	SUBJECT	27, 74	/	AB
							DEPT-ACADEMIC NO MM			
4	DEPT-GENERAL	DEPT	27,	48	/	AA	SUBJECT	27, 74	/	AB
							DEPT-GENERAL NO MM			
5	DEPT-TEACHER	DEPT	27,	48	/	AA	TEACHER	70, 48	/	AA
							DEPT-TEACHER NPO OM			
6	IX-DEPT-NAME	SYSTEM (SR7)					DEPT	27, 48	/	AA
							IX-DEPT-NAME N MA			
7	IX-STUD-LNAME	SYSTEM (SR7)					STUDENT	70, 12	/	AA
							IX-STUD-LNAME N MA			

Exhibit 4.10: Tailored Cross-Reference Report with Modified Set Descriptions

4.4 Example 4

Example 4 shows how the user positioned four record blocks--ACTIVITY, STUDENT, SCHEDULE, and DEPT--in the data structure diagram (see Exhibit 4.12).

Why record blocks were manually positioned in the diagram--The user was mainly concerned with the positions of four key record blocks in the data structure diagram. The user wanted to specify the locations of the four record blocks in the diagram and allow CA-IDMS/Schema Mapper to automatically LOCATE the rest of the record blocks.

Parameters Entered--The user could have edited the LOCATE statements in the Transfer File produced with the default diagram and then use the Transfer File as input. But in this case, the user would have to edit 16 LOCATE statements in the Transfer File.

Instead, the user simply added one LOCATE statement to the original Parameter Statements File to specify the position for each of the four record blocks (see Exhibit 4.11). This action LOCATED the four record blocks in the desired positions and allowed CA-IDMS/Schema Mapper to position the rest of the record blocks automatically.

```
PROCESS=SUBSCHEMA, SCHEMA=TSDSCHM1, SUBSCHEMA=TSDSUBS1,
      DICTNAME=SMAP
```

LOCATE ACTIVITY					
LOCATE STUDENT			DOWN	1	FROM ACTIVITY
LOCATE SCHEDULE			DOWN	1	FROM STUDENT
LOCATE DEPT	LEFT	1	DOWN	2	FROM STUDENT

Exhibit 4.11: Example 4--Parameters Entered

4.4 Example 4

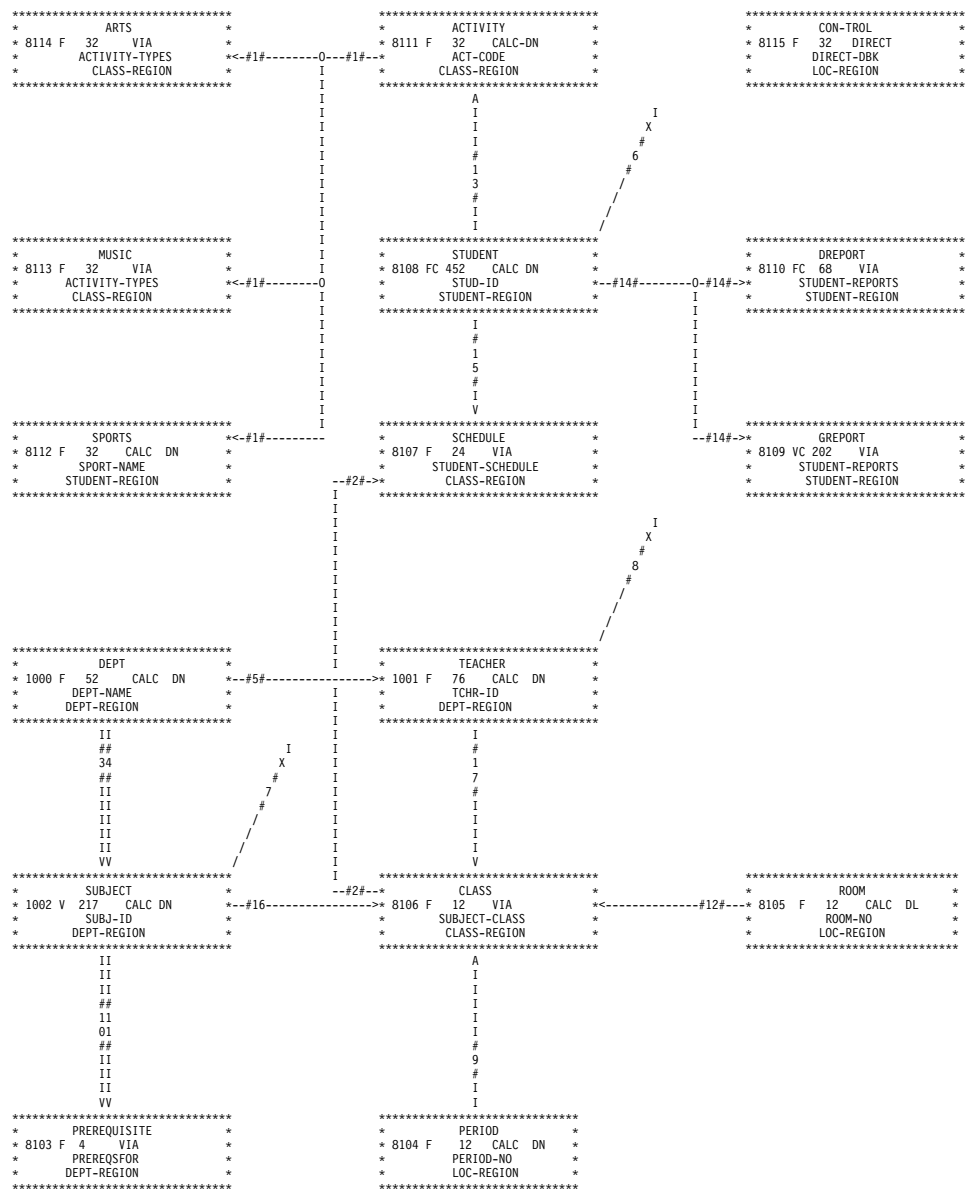


Exhibit 4.12: Tailored CA-IDMS/Schema Mapper Data Structure Diagram with Manually Positioned Record Blocks

4.5 Example 5

Here, the user had a schema containing multiple CALC and sort keys. The purpose of this example is just to show what CA-IDMS/Schema Mapper does with multiple CALC and sort keys; it does not demonstrate an additional use of parameter statements.

Parameters Entered--The user entered a single PROCESS statement to request that CA-IDMS/Schema Mapper produce the data structure Diagram and Cross-Reference Report automatically and according to default values. See Exhibit 4.13.

Output Produced--Exhibits 4.14 and 4.15 show the data structure Diagram and the Cross-Reference Report. Note that every record block has enough space to accommodate the record type with the most CALC keys. Also note that every set description has enough space to accommodate the set with the most sort keys.

```
PROCESS=SUBSCHEMA, SCHEMA=DS16MULT, SUBSCHEMA=DB16MULTI, VERSION=1,  
      DICTNAME=TKIT
```

Exhibit 4.13: Example 5--Parameters Entered

CA-IDMS/SCHEMA MAPPER:		mm/dd/yy hh:mm:ss	SUBSCHEMA DBRR1016, SCHEMA DSRR1016, VERSION 1, DICTNAME=TKIT, DBNAME=	
RECORD CROSS-REFERENCE LISTING				
RECORD NAME	LOCATION		/ PAGE	

ACTIVITY	94,	27	/ AA	
ARTS	150,	27	/ BA	
CLASS	94,	74	/ AB	
CON-TROL	1,	12	/ AA	
DEPT	39,	48	/ AA	
DREPORT	39,	12	/ AA	
GREPORT	39,	27	/ AA	
MUSIC	150,	48	/ BA	
PERIOD	94,	95	/ AB	
PREREQ	39,	95	/ AB	
CA-IDMS/SCHEMA MAPPER:		MM/DD/YY HH:MM:SS	SUBSCHEMA DBRR1016, SCHEMA DSRR1016, VERSION 1, DICTNAME=TKIT, DBNAME=	
SET/INDEX CROSS-REFERENCE LISTING				
SET NUMBER	SET OR INDEX NAME	OWNER	LOCATION	/ PAGE

1	ACTIVITY-TYPES	ACTIVITY	94, 27	/ AA
		ARTS	150, 27	/ BA
		ACTIVITY-TYPES		
		NP OA		
		LAST		
		MUSIC	150, 48	/ BA
		ACTIVITY-TYPES		
		NP OA		
		LAST		
		SPORTS	150, 74	/ BB
		ACTIVITY-TYPES		
		NPO OM		
		LAST		
2	CLASS-SCHEDULE	CLASS	94, 74	/ AB
		SCHEDULE	150, 12	/ BA
		CLASS-SCHEDULE		
		NPO MA		
		ASC ROSTER-NO DL		
3	DEPT-ACADEMIC	DEPT	39, 48,	/ AA
		SUBJECT	39, 74	/ AB
		DEPT-ACADEMIC		
		NO MM		
		NEXT		
4	DEPT-GENERAL	DEPT	39, 48	/ AA
		SUBJECT	39, 74	/ AB
		DEPT-GENERAL		
		NO MM		
		NEXT		
5	DEPT-TEACHER	DEPT	39, 48	/ AA
		TEACHER	94, 48	/ AA
		DEPT-TEACHER		
		NPO OM		
		ASC TCHR-NAME DL		
		ASC TCHR-ID DL		
		ASC TCHR-SUBJECT DL		
6	IX-DEPT-NAME	SYSTEM (SR7)	DEPT	39, 48 / AA
		IX-DEPT-NAME		
		N MA MODE=INDEX		
		ASC DEPT-NAME DF		
7	IX-STUD-LNAME	SYSTEM (SR7)	STUDENT	94, 12 / AA
		IX-STUD-LNAME		
		N MA MODE=INDEX		
		DESC DBKEY DL		

Exhibit 4.15: Default Cross-Reference Report with Multiple Sort Keys

Chapter 5. Operations

5.1 System Requirements	5-4
5.1.1 Storage Requirements	5-4
5.2 OS/390 Environments	5-5
5.2.1 OS/390 JCL	5-5
5.2.2 Key to OS/390 JCL	5-5
5.3 VSE/ESA Environments	5-8
5.3.1 VSE/ESA File Assignments	5-8
5.3.2 VSE/ESA JCL	5-8
5.3.3 Key to VSE/ESA JCL	5-10
5.4 VM/ESA Environments	5-12
5.4.1 VM/ESA EXEC	5-12
5.4.2 Key to VM/ESA EXEC	5-13

This chapter covers CA-IDMS/Schema Mapper operations, including system requirements and storage requirements.

5.1 System Requirements

CA-IDMS/Schema Mapper runs under OS/390, VSE/ESA, and VM/ESA operating systems and CA-IDMS 15.0.

5.1.1 Storage Requirements

CA-IDMS/Schema Mapper needs 250K for program storage. If you are mapping a small schema/subschema (with less than 40 records), you need an additional 700K; a medium-sized schema/subschema (40 to 100 records) requires an additional 1700K.

5.2 OS/390 Environments

5.2.1 OS/390 JCL

The JCL for OS/390 execution is contained in Target or Distribution source library member USSEXEC and is shown below.

```
//SCHMAPR EXEC  PGM=USSCMPR,TIME=10,REGION=#K
//STEPLIB      DD DSN=your.loadlib,DISP=SHR
//             DD DSN=idms.loadlib,DISP=SHR
//*   JES2 CONTROL TO SUPPRESS PAGE EJECT ON DIAGRAM FILE
/*OUTPUT DIAG LINECT=0
//*   JES3 CONTROL TO SUPPRESS PAGE EJECT ON DIAGRAM FILE
/*FORMAT PR,DDNAME=DIAGRAM,OVFL=OFF
//SYSCTL       DD DSN=idms.sysctl,DISP=SHR
//dictdb       DD DSN=idms.dictionary,DISP=SHR
//SYSJRN       DD DUMMY
//*   DIAGNOSTIC AID FILES
//SYSPRINT     DD SYSOUT=a
//SYSTEM       DD SYSOUT=a
//OUTPUT       DD SYSOUT=a
//*   OUTPUT FILES
//*   TRANSFER FILE
//SYSPCH       DD DSN=schemapr.parm.transfer.file,
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3280),
//             SPACE=(TRK,(1,1)),UNIT=disk,DISP=(NEW,CATLG)
//*   AUDIT REPORT FILE
//SYSLS       DD SYSOUT=a
//*   RECORD AND SET/INDEX CROSS REFERENCE FILE
//XREF         DD SYSOUT=a
//*   DIAGRAM (MAP) FILE
//DIAGRAM      DD SYSOUT=ds
//*   TEMPORARY WORK FILE
//SCRATCH      DD SPACE=(TRK,(1,1)),UNIT=disk,
//             DCB=(RECFM=F,LRECL=80)
//*   PARAMETER INPUT FILE
//STATFILE     DD DSN=schemapr.stats.input.file,DISP=SHR
//SYSIPT       DD DSN=schemapr.parm.input.file,DISP=SHR
/*
//SYSIDMS      DD *
              SYSIDMS PARAMETERS...
/*
```

5.2.2 Key to OS/390 JCL

- **#K** — The amount of memory needed to execute CA-IDMS/Schema Mapper, in kilobytes. CA-IDMS/Schema Mapper needs about 250K for program storage. If you are mapping a small schema/subschema (with less than 40 records), you need an additional 700K; a medium-sized schema/subschema (40 to 100 records) requires an additional 1700K.
- **your.loadlib** — The dataset name of the library into which the CA-IDMS/Schema Mapper load modules were copied when you downloaded the tape (during Step 1 of OS/390 installation).

- **/*OUTPUT (JES2)** and **/*FORMAT (JES3)** — These statements are needed only in OS/390 environments for proper handling of page ejects in the DIAGRAM output. With these statements, the DIAGRAM will be in a separate SYSOUT dataset from the rest of the output.
- **idms.loadlib** — The dataset name of the library in which your CA-IDMS modules reside.
- **idms.sysctl** — The dataset name of the CA-IDMS system control file. This is used to request central version (CV) operation.
- **dictdb** — The DDNAME of the CA-IDMS dictionary being accessed. DICTDB is the default. Your site may have changed the DDNAME in the dictionary DMCL. This statement is needed if running local mode.
- **idms.dictionary** — The dataset name of the dictionary that contains the schema or subschema information.
- **SYSJRNL** — This statement is needed if running local mode.
- **a** — The appropriate SYSOUT class for your installation.
- **schemapr.parm** and **transfer.file** — The dataset name of the Transfer File (output file) to which parameter statements are to be written.
- **ds** — For OS/390 JES2 environments, replace with (**a**.,DIAG). Otherwise, use **a**. See **a** above for both.
- **disk** — The unit address of the disk drive you want to use for the CA-IDMS/Schema Mapper execution file or a generic assignment to indicate a disk drive allocation.
- **schemapr.stats** and **input.file** — The dataset name of the statistics file that was created by CA-IDMS/DB ANALYZER
- **schemapr.parm** and **input.file** — The dataset name of the file containing the parameter statements to be used as input. This can be either a Transfer File created by CA-IDMS/Schema Mapper during a previous run or a user-created file.

You must place a JOB card at the beginning of this JCL.

CA-IDMS/Schema Mapper writes the SCRATCH file and later reads it.

Instead of defining an input file (that contains parameter statements) with the SYSIPT statement in the preceding JCL, you can use this SYSIPT statement:

```
//SYSIPT DD *
```

followed by the parameter statements to be used as input.

The statistics file is needed only if you request one of the statistics fields in your output. If you do not request one of the statistics fields in your output, STATFILE can be DUMMY or undefined.

CA-IDMS/Schema Mapper treats a column of the diagram as a single logical page with no forced page ejects. In an OS/390/JES environment, a page eject is usually created automatically for any SYSOUT file when the data does not include a page eject. This prevents CA-IDMS/Schema Mapper from writing more than 60 lines on a page. You need to include statements to suppress automatic page ejects.

The various releases of JES use different statements to suppress the automatic page ejects. The sample JCL includes both JES2 and JES3 statements for suppressing page ejects. Before using these statements, you need to check that their formats are appropriate for your environment.

You must use the following SYSIDMS statement:

```
//SYSIDMS DD*
```

followed by SYSIDMS parameters. This file is always required for the CA-IDMS physical environment parameter input to be read. For instance, DMCL=xxxxxxx must be specified, where xxxxxxxx is the name of the specific DMCL. See the *CA-IDMS 15.0. User Guide* for more information.

5.3 VSE/ESA Environments

5.3.1 VSE/ESA File Assignments

Even if you use a storage management tool such as CA-DYNAM, CA-IDMS/Schema Mapper requires an ASSGN statement for every file except SORTWK*nn*. This ASSGN is required because CA-IDMS/Schema Mapper has its own device-independent support which dynamically builds a DTF based on the device type indicated by the ASSGN. Unless the ASSIGN specifies VSAM or BDAM, the file may be defined with either DLBL or TLBL.

5.3.2 VSE/ESA JCL

The JCL for VSE/ESA execution is contained in TOOLJCL library member USSEXEC.S (VSE/ESA), and is shown below:

```

// UPSI      nnnnnnnn          FOR CENTRAL VERSION EXECUTION
// OPTION PARTDUMP              DUMP REQUESTED FOR ERROR ABORTS
// ASSGN SYSIPT,SYSRDR          PARAMETER FILE(INPUT)
// ASSGN SYSPCH,00D             FOR TRANSFER FILE(OUTPUT)
// ASSGN SYSLST,00E            AUDIT REPORT FILE(OUTPUT)
// ASSGN SYS010,00E            DIAGRAM FILE(OUTPUT)
// DLBL SCRATCH,'schemapr.work.file',0,SD
// EXTENT SYS012,volser,,0,starttrack,#tracks
// ASSGN SYS012,DISK,VOL=volser SCRATCH/WORK FILE(OUTPUT/INPUT)
// DLBL STATFIL,'dbnlyzr.stats.file',0,SD
// ASSGN SYS013,DISK,VOL=volser OPTIONAL STATISTICS FILE
// EXTENT SYS013,volser,,0,starttrack,#tracks
// ASSGN SYS014,00E            XREF REPORT FILE(OUTPUT)
* *****
* If running in LOCAL mode, include dataset containing the DDLDML area
* of the dictionary containing the description of the desired subschema
* *****
// DLBL DICTDB,'your.dict.ddldml'
// EXTENT SYSnnn,volser,,starttrack,#tracks
// ASSGN SYSnnn,DISK,VOL=volser,SHR
* *****
* -----LOAD LIBRARIES-----
// DLBL DBMS,'your.loadlib' CA/IDMS-TOOLS INSTALLED IN
// EXTENT ,volser
// DLBL IDMS,'idms.loadlib' CA/IDMS INSTALLED
// EXTENT ,volser
* *****
/* For VSE/ESA use the following statement
// LIBDEF CL,SEARCH=(IDMS,DBMS)
* *****
/* For VSE/ESA use the following statement
// LIBDEF PHASE,SEARCH=(DBMS.sublib,IDMS.sublib)
* *****
* FOR CV runs specify:
// DLBL SYSCTL,'your.sysctl.file'
// EXTENT SYSnnn,volser,,starttrack,#tracks
// ASSGN SYSnnn,DISK,VOL=volser,SHR
* *****
// SYSIDMS,'#SYSIPT',0,SD
// EXEC USSCMPR,SIZE=(USSCMPR,512K) EXECUTE Schema Mapper
/*
* *****Schema Mapper SYNTAX *****
PROCESS=----- PARAMETER CONTROL STATEMENTS
/*
* SYSIDMS Parameter Statements for 15.0 IDMS
*
* 15.0 SYSIDMS parameters.
* For Local Mode specify :
*
DMCL=dmc1-name,LOCAL=ON,JOURNAL=OFF
*
* For CV runs specify :
*
DMCL=dmc1-name,LOCAL=OFF,JOURNAL=OFF
/*

```

5.3.3 Key to VSE/ESA JCL

- **nnnnnnnn** — The UPSI byte settings appropriate to the IDMSOPTI module, which you have linked with CA-IDMS/Schema Mapper, and to the method you are using to access CA-IDMS--Local Mode or central version (CV).
- **00D** — The unit address of your punch device.
- **00E** — The unit address of your printer device.
- **schemapr.work.** and **file** — The name of the scratch/work file that is used for Parameter File input and Transfer File output.
- **volser** — The volume serial number or generic assignment of the disk volume on which the file, specified in the previous statement, resides.
- **starttrack** — Available disk extent on the work disk volume (1 to 6 digits).
- **#tracks** — Sufficient space for the input Parameter File (1 to 6 digits). Ten tracks is usually sufficient space.
- **your.dict.ddldml** — The data set containing the DDLDML area of the dictionary containing the description of the desired subschema.
- **your.loadlib** — The data set name of the core image library into which you downloaded CA-IDMS/Schema Mapper.
- **idms.loadlib** — The data set name of the core image library in which your DMCL and subschema reside.
- **DBMS.sublib** and **IDMS.sublib** — The sublibrary name of the VSE/ESA library specified in the previous file name.
- **your.sysctl.file** — The file name of your SYSCTL file.
- **SYSnnn** — The programmer logical unit for the file specified in the previous DLBL statement.
- **dmcl-name** — Specify the name of the DMCL that should be used when accessing the dictionary.

Note: When running in local mode, the job control for the dictionary can be defined in standard labels or partition standard labels instead of in stream.

The size parameter in this JCL specifies the PPA space needed for CA-IDMS/Schema Mapper and CA-IDMS program storage. The remainder of the partition will be used for GETVIS storage.

The amount of GETVIS storage required depends on the size of the schema/subschema you are mapping. If you are mapping a small schema or subschema (with less than 40 records), you need about 700K; a medium-sized schema or subschema (40 to 100 records) requires about 1700K.

The files in this JCL can be ASSGned to any device type on the system that has or that can receive the data for CA-IDMS/Schema Mapper to operate. (There is full device independence.) If you use disk, DLBL and EXTENT JCL must be provided

either in the above JCL or in standard labels. If you use labeled tape, TLBL JCL must be included in the above JCL. Here are the DLBL/TLBL names for the corresponding file ASSGN control statements:

DIAGRAM SYS010
SCRATCH SYS012
STATFILE SYS013
XREF SYS014

These are the names for tape; disk names will be truncated to the first seven characters as necessary. Also, CA-IDMS/Schema Mapper writes the SCRATCH file; then reads it.

If you do not want a dump when a CA-IDMS/Schema Mapper job terminates abnormally, you can include a // OPTION NODUMP control statement in the JCL. If you do not use // OPTION NODUMP or if you choose to include // OPTION DUMP or // OPTION PARTDUMP, a dump will be produced for an abend in addition to any diagnostic messages from CA-IDMS/Schema Mapper and the Pascal run-time modules.

5.4 VM/ESA Environments

5.4.1 VM/ESA EXEC

A sample VM/ESA EXEC for executing CA-IDMS/Schema Mapper appears below. Variables (*italics*) are explained in the key following the EXEC.

Note: Separate load modules should be generated for CV and local mode execution. See the *CA-IDMS Tools Installation and Maintenance Guide* for details.

```
/* */
TRACE OFF; SIGNAL ON ERROR
/*
CA_LOADLIB_FN      = 'yourlib'
IDMS_LOADLIB_FN    = 'idmslib'
/*
/* Link and access the Minidisks containing the required librerie(s) */
/*
'CP SPOOL PRINTER NOCONT CLOSE'
'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
'GLOBAL LOADLIB ' CA_LOADLIB_FN IDMS_LOADLIB_FN
/*
/* Product specific files.
/*
'FILEDEF SYSTEM PRINTER'
'FILEDEF SYSPRINT PRINTER'
'FILEDEF SYSUDUMP PRINTER'
'FILEDEF OUTPUT PRINTER'
'FILEDEF SYSLST PRINTER'
'FILEDEF XREF PRINTER'
'FILEDEF DIAGRAM PRINTER'
'FILEDEF SCRATCH PRINTER'
'FILEDEF SYSPCH DISK USSEEXEC SYSPCH A'
'FILEDEF SYSIPT DISK USSEEXEC SYSIPT A'
/* SYSCTL file - remove for local mode operation
'FILEDEF SYSCTL DISK sysctl ft fm'
/*
```

```

/* If you are running in Local Mode remove the comments from the */
/* next 4 Dictionary and Journal FILEDEF statements. */
/*
'FILEDEF DICTDB DISK dictfn dictft dictfm ( dcb extent
'FILEDEF DMSGDB DISK dmsgfn dmsgft dmsgfm ( dcb extent
'FILEDEF DLODDB DISK dlodfn dlodft dlodfm ( dcb extent
'FILEDEF SYSJRNL DUMMY'
*/
/*
/* CA-IDMS/DB ANALYZER Statistic File. */
/* Include the following file definition if CA-IDMS/DB ANALYZER */
/* have been specified for CA-IDMS/Schema Mapper. */
/*
'FILEDEF STATFILE DISK statfn statft statfm'
/*
/* You must create a file 'SYSIDMS INPUT A' containing the SYSIDMS */
/* parameters you use to specify your runtime environment. */
/*
'FILEDEF SYSIDMS DISK SYSIDMS INPUT A'
/*
SIGNAL OFF ERROR
SAY 'STARTING CA-IDMS/Schema Mapper'
'EXECOS OSRUN USSCMPR'
USSCMPR_RC = RC
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME USSEXC LISTING'
'CP SPOOL PRINTER OFF'
SAY 'USSEXC FINISHED WITH A RETURN CODE OF' USSCMPR_RC
'GLOBAL LOADLIB'
'FILEDEF * CLEAR'
EXIT USSCMPR_RC
/*
/***** */
ERROR:
/***** */
ERROR_RC = RC
TRACE OFF; SIGNAL OFF ERROR
/*
SAY 'NON-ZERO RETURN CODE ENCOUNTERED IN EXEC AT LINE' SIGL
/*
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME USSCMPR LISTING'
'CP SPOOL PRINTER OFF'
'GLOBAL LOADLIB'
'FILEDEF * CLEAR'
EXIT ERROR_RC
/*

```

5.4.2 Key to VM/ESA EXEC

- **yourlib** — The file name of the load library into which you downloaded CA-IDMS/Schema Mapper.
- **idmslib** — The file name of the load library containing your CA-IDMS load modules.
- **dictfn dictft dictfm** — The file name, file type, and file mode of the CA-IDMS Data Dictionary Directory area.
- **sysctl fm ft** — The file name, file type, and file mode of the CA-IDMS SYSCTL file.

- **dmsgfn dmsgft dmsgfm** The file name, file type, and file mode of the CA-IDMS Data Dictionary Message area.
- **dlodfn dlodft dlodfm** — The file name, file type, and file mode of the CA-IDMS Data Dictionary Load area.
- **dcb extent** — The data control block (dcb) and extent information required for the dictionary files.
- **ststfn ststft ststfm** — The file name, file type, and file mode of the statistics file created by CA-IDMS/DB ANALYZER.
- **USSEEXEC SYSPCH A** — The file name, file type, and file mode of the transfer file (output file) to which parameter statements are written.
- **USSEEXEC SYSIPT A** — The file name, file type, and file mode of the file containing the input parameter statements.

The FILEDEFs for DICTDB, DMSGDB, DLODDb, and SYSJRNl can be removed when running under CV.

Chapter 6. Messages

This chapter describes the three types of messages generated by the CA-IDMS/Schema Mapper Audit Report: informative, warning, and error. This chapter lists the codes for each message, gives reasons for their occurrences, and suggests remedial actions when appropriate. The severity codes are explained below.

Informative--Prefixed by an *Innn* message code. Informative messages report processing starts, stops, and parameter information.

Warning--Prefixed by an *Wnnn* message code. Warning messages report conflicting parameter data. When CA-IDMS/Schema Mapper reports a warning messages, it performs corrective actions to continue processing.

Error--Prefixed by an *Ennn* message code. Error messages report erroneous and conflicting parameter data that causes processing to terminate. CA-IDMS/Schema Mapper continues, however, to scan the parameter statements for other syntactical errors.

Note: If CA-IDMS/Schema Mapper encounters a blank field to be printed in a message, it prints the field as four question marks.

I001 Schema Mapper STARTED mm/dd/yy hhmmss

Reason: Displays the date in month, day, and year and the time in hour, minute, and second that CA-IDMS/Schema Mapper started processing.

Action: None.

I002 PARAMETER INPUT.....parameter statement

Reason: Displays the parameter statements for this execution of CA-IDMS/Schema Mapper.

Action: None.

I003 Schema Mapper ENDED mm/dd/yy hhmmss

Reason: Displays the date in month, day, and year and the time in hour, minute, and second that CA-IDMS/Schema Mapper ended processing.

Action: None.

I004 MAXIMUM X,Y COORDINATES OF DIAGRAM ARE x,y

Reason: Display of the maximum coordinates for which space was allocated.

Action: None.

I005 DIAGRAM GENERATION SUPPRESSED BY ERRORS. SYNTAX ANALYSIS CONTINUES.

Reason: The diagram cannot be generated because of fatal errors. CA-IDMS/Schema Mapper continues to scan the parameter statements for syntactical errors.

Action: Review error messages and resolve errors after CA-IDMS/Schema Mapper finishes processing.

I006 GENERATED STATEMENT.....parameter statement

Reason: Displays a parameter statement that CA-IDMS/Schema Mapper has generated. Generally, this is a LOCATE statement for a record block that you did not manually LOCATE.

Action: None.

I007 THIS SCHEMA WAS COMPILED USING IDMS VERSION value ON mm/dd/yy

Reason: Displays the version of CA-IDMS used to compile the schema and the date of the schema compile (in month, day, and year). (If the DATE FIELD from the SCHEMA DESCRIPTION has been filled in, that date is displayed instead of the compile date.)

Action: None.

I008 THIS SUBSCHEMA WAS COMPILED ON mm/dd/yy

Reason: Displays the date of the subschema compile (in month, day, and year). The version of CA-IDMS used to compile a subschema is not available in the dictionary/directory. (If the DATE FIELD from the SUBSCHEMA DESCRIPTION has been filled in, that date is displayed instead of the compile date.)

Action: None.

I009 USING SCHEMA VERSION version-number

Reason: Displays the version of the schema.

Action: None.

I010 Schema Mapper IS RUNNING UNDER IDMS VERSION version-number

Reason: Displays the version of CA-IDMS under which CA-IDMS/Schema Mapper is running.

Action: None.

I011 SKIPPING KEY NUMBER number IN name kind . . .

Reason: RECLINE or SETLINE statements did not include this key in the Data Structure Diagram or Cross-Reference Report. Number corresponds to the implied subscript in LOC-CTRL or SET-ORDER. Kind is RECORD, SET, or INDEX. Name is the name of the record, set, or index.

Action: None.

I012 . . .KEY NAME IS key-name

Reason: This message is a continuation of I011. The name of the key referred to in the preceding I011 is indicated by key-name in this message.

Action: None.

I013 THIS DICTIONARY DOES NOT CONTAIN ANY SCHEMA RECORDS

Reason: CA-IDMS/Schema Mapper did not find any schema records in the requested dictionary.

Action: Add schemas to this dictionary or use a different dictionary.

I014 SCHEMA NAME IS name VERSION IS version-number

Reason: CA-IDMS/Schema Mapper uses this message when listing the contents of a dictionary.

Action: None.

I015 SUBSCHEMA NAME IS name

Reason: CA-IDMS/Schema Mapper uses this message when listing the contents of a dictionary.

Action: None.

I016 (THIS SCHEMA DOES NOT CONTAIN ANY SUBSCHEMAS)

Reason: CA-IDMS/Schema Mapper did not find any subschemas defined for the current schema.

Action: None.

W001 MULTIPLE LOCATE STATEMENTS FOR RECORD record-name

Reason: More than one LOCATE statement is used for the designated record. CA-IDMS/Schema Mapper processes the first LOCATE and ignores the second statement.

Action: Verify that the record is located where you want it. Delete the extra statement.

W002

Reason: This message is reserved for technical support purposes.

Action: Contact Computer Associates Technical Support.

W003 SPACE REQUESTED FOR RECORD record-name ALREADY ALLOCATED

Reason: A LOCATE statement requests space already allocated to another record through a previous LOCATE. The LOCATE statement is skipped, which can create a ripple effect in which subsequent LOCATES are flagged with the W005 message number. CA-IDMS/Schema Mapper places the first record specified in the LOCATE statement in the diagram after all LOCATE statements are processed.

Action: Verify placement of the records. Revise or delete LOCATES as needed.

W004 RECORD ID number FOR name IS OUT OF RANGE

Reason: The record ID number for the record indicated is either less than 1 or greater than 9999. A record ID value of 0 is used for this record.

Action: Check the record ID value in the CA-IDMS dictionary. The value is created using the Record Description paragraph in the Schema Data Description Language. The value is stored in the SR-ID-113 element of the SRCD-113 record in the dictionary.

W005 THE 'FROM' RECORD record-name HAS NOT YET BEEN LOCATED

Reason: The second record name in a LOCATE statement is used as a reference. It must be LOCATED before the current LOCATE can be processed. A LOCATE statement is missing, or the LOCATE statements are out of sequence. The current LOCATE statement is skipped, which can create a ripple effect in which subsequent LOCATES are flagged with the same warning. CA-IDMS/Schema Mapper places the first record of the LOCATE statement in the diagram after all LOCATE statements are processed.

Action: Check to see that a LOCATE statement is not missing or out of sequence.

W006

Reason: This message is reserved for technical support purposes.

Action: Contact Computer Associates Technical Support.

W007 DICTIONARY CONTAINED INCONSISTENT SET ORDER AND SET CONNECT VALUES FOR set-name CASE number

Reason: For the set or index indicated (by set-name), the set connect value obtained from the CA-IDMS dictionary is not consistent with the set order value obtained from the dictionary. The case number in the message indicates what has occurred:

CASE 0 The set order is 0 (last or prior) and the set connect is not 0 or 1.

CASE 1 The set order is 1 (sorted) and the set connect is not 0.

CASE 2 The set order is 2 (first or next) and the set connect is not 0 or 1.

Action: Check your dictionary for the values of these two elements. Set order is stored in the SET-ORD-046 element of the SOR-046 record for both schemas and subschemas. Set connect is stored in the ORD-052 element of the SMR-052 record for schemas and in the ORD-068 element of the SSMR-068 record for subschemas.

The meaning of set connect values are:

0 = linked on owner (for first, last, or sorted)

1 = connect on current of set (for next, prior)

W008 DICTIONARY CONTAINED INAPPROPRIATE SORT SEQUENCE VALUE FOR set-name

Reason: For the set or index indicated (by set-name), the sort sequence value was not 0 or 1.

Action: Check your dictionary for the value of this element. Sort sequence is stored in SORT-054 for schemas and SORT-070 for subschemas.

W009 DICTIONARY CONTAINED INAPPROPRIATE DUPLICATES VALUE FOR set-name

Reason: For the set or index indicated (by set-name), the duplicates value was not -1, 0, 1, or 2.

Action: Check your dictionary for the value of this element. Duplicates is stored in DUP-052 for schemas and DUP-068 for subschemas.

W010 DICTIONARY CONTAINED INAPPROPRIATE SORT ORDER VALUE FOR set-name

Reason: For the set or index indicated (by set-name), the sort order value was not 0, 1, or 2.

Action: Check your dictionary for the value of this element. Sort order is stored in SORT-046 for both schemas and subschemas.

W011 DICTIONARY CONTAINED INAPPROPRIATE MEMBERSHIP OPTIONS VALUE FOR set-name CASE number

Reason: For the set or index indicated (by set-name), the membership options value was inappropriate. The case number in the message indicates what has occurred:

CASE 1 The membership options value is -1.

CASE 2 The membership options value is not -1, 0, 1, 2, or 3.

Action: Check your dictionary for the value of this element. Membership options is stored in MR-CNTRL-052 for schemas and MCTL-068 for subschemas.

W012 LOCATION REQUESTED FOR RECORD record-name IS OUTSIDE LAYOUT WORK SPACE

Reason: A LOCATE statement requests placement of a record block in a location that is outside of the work space used by CA-IDMS/Schema Mapper to construct the diagram. The LOCATE statement is skipped, which can create a ripple effect in which subsequent LOCATEs are flagged with the W005 message number. CA-IDMS/Schema Mapper places the first record specified in the LOCATE statement in the diagram after all LOCATE statements are processed.

Action: Verify placement of the records. Change the displacement value(s) in the LOCATE statement.

W013 SET/INDEX/RECORD name HAS BEEN PLACED OUTSIDE OF THE DIAGRAM PAGE LIMITS

Reason: CA-IDMS/Schema Mapper allows only 26 pages across and 26 pages down for the diagram. The indicated item is at least partially outside of this limit. The part of the item that is outside does not appear in the diagram. By the time CA-IDMS/Schema Mapper detected this problem, the location of the item could not be changed.

Action: Use LOCATE statements to change either the location of the indicated record block or the locations of the record blocks that participate in the indicated set or index.

W014 MINIMUM FRAGMENT SPECIFIED FOR NON-COMPRESSED FIXED RECORD record-name

Reason: The CA-IDMS dictionary entry specified a minimum fragment for the indicated record, but the record is a non-compressed fixed record. The LENGTH-MODE shown for the record in the diagram may be incorrect.

Action: Check the dictionary for the schema. You may find an error in the definition of this record. For example, if you wanted a length-mode of FC, you may have forgotten to request COMPRESS/DECOMPRESS.

W015 kind-of-name HAS ALREADY BEEN INCLUDED IN THE DIAGRAM

Reason: CA-IDMS/Schema Mapper has already included the indicated name in the diagram. This was done when processing an earlier INCLUDE statement. The kind-of-

name is AREA, RECORD, SET, or INDEX. CA-IDMS/Schema Mapper ignores the INCLUDE statement preceding this message.

Action: Remove the redundant INCLUDE statement from the input file (SYSIPT).

W016 LOCATE STATEMENT IGNORED BECAUSE RECORD *rec-name* NOT INCLUDED

Reason: INCLUDE statements have been used, and the indicated record was not included in the diagram. Therefore, a LOCATE statement for this record has no meaning. The LOCATE statement preceding this message is ignored.

Action: Remove the irrelevant LOCATE statement from the input file (SYSIPT).

W017 RECORD *rec-name* HAS TOO MANY SET CONNECTIONS

Reason: The indicated record participates as an owner or a member in more than 128 sets. CA-IDMS/Schema Mapper does not draw some of the set connections in the diagram. (See message W018.)

Action: Notify Computer Associates Technical Support.

W018 NO ROOM FOR SET *set-name* IN THE DIAGRAM

Reason: CA-IDMS/Schema Mapper has previously used message W017 for a record that participates in this set.

Action: Notify Computer Associates Technical Support.

W019 DICTIONARY CONTAINED INAPPROPRIATE SET MODE VALUE FOR *set-name*

Reason: For the set or index indicated by *set-name*, the set mode value was inappropriate. Legal values for set-mode are 13, 15, 21, and 32 (decimal).

Action: Check your dictionary for the value of this element. Set mode is stored in SET-MODE-046 for both schemas and subschemas.

W020 SET-MODE NOT AVAILABLE UNDER THIS VERSION OF IDMS

Reason: The SET-MODE field was requested in a SETLINE statement, but a set mode of INDEX is not available in the version of CA-IDMS under which CA-IDMS/Schema Mapper is running. (See message I010 in the Audit Report.) CA-IDMS/Schema Mapper uses a blank SET-MODE value and continues processing.

Action: Do not use the SET-MODE field with this version of CA-IDMS.

W021 MULTIPLE KEYS NOT AVAILABLE UNDER THIS VERSION OF IDMS

Reason: A subscript was used with the LOC-CTRL field in a RECLINE statement or the SET-ORDER field in a SETLINE statement. Subscripts on these fields are used only by the multiple CALC, sort, and index key feature that is not available in the version of CA-IDMS under which CA-IDMS/Schema Mapper is running. (See message I010 in the Audit Report.) The first LOC-CTRL or SET-ORDER field has its normal value. (The use of LOC-CTRL(1) and SET-ORDER(1) is always allowed and does not trigger this message.) CA-IDMS/Schema Mapper uses a blank value for any additional LOC-CTRL or SET-ORDER fields and continues processing.

Action: Do not use LOC-CTRL or SET-ORDER subscripts with this version of CA-IDMS.

W023 THE RECLINE STATEMENT IS OUTDATED. USE DRECLINE INSTEAD

Reason: You can place record fields in both the diagram and the cross-reference listing by using DRECLINE and XRECLINE statements respectively. Because these statements replace the RECLINE statement, any use of the RECLINE statement is interpreted as a DRECLINE statement by CA-IDMS/Schema Mapper.

Action: Translate RECLINE statements to DRECLINE statements.

W024 THE SETLINE STATEMENT IS OUTDATED. USE XSETLINE INSTEAD

Reason: You can place set fields in both the diagram and the cross-reference listing by using DSETLINE and XSETLINE statements respectively. Because these statements replace the SETLINE statement, any use of the SETLINE statement is interpreted as an XSETLINE statement by CA-IDMS/Schema Mapper.

Action: Translate SETLINE statements to XSETLINE statements.

W025 STATISTICS INPUT FILE IS EMPTY. CHECK SPELLING OF NAME GIVEN FOR STATFILE.

Reason: Fields defined for DRECLINE, XRECLINE, DSETLINE, or XSETLINE require information from the STATFILE. The JCL does not reference a valid STATFILE.

Action: Change the xxxxLINE parameters or correct the JCL.

W026 STATISTICS ARE NOT AVAILABLE FOR set-name.

Reason: Fields defined for DSETLINE or XSETLINE require information from the STATFILE. The schema/subschema being mapped contains set-name, but STATFILE does not contain statistics for that set.

Action: Change the XSETLINE parameters or supply a proper STATFILE.

W027 STATISTICS ARE NOT AVAILABLE FOR MEMBER NAME member-name OF set name

Reason: Fields defined for DRECLINE, XRECLINE, DSETLINE, or XSETLINE require information from the STATFILE. The schema/subschema being mapped contains set-name and member-name, but STATFILE does not contain statistics for that set.

Action: Change the XSETLINE parameters or supply a proper STATFILE.

W028 RECORD ID IS DIFFERENT IN SCHEMA AND DATABASE FOR MEMBER member-name OF SET set name

Reason: Fields defined for DRECLINE, XRECLINE, DSETLINE, or XSETLINE require information from the STATFILE. The schema/subschema being mapped contains set-name and member-name, but STATFILE contains a record ID for member-name that is different than the record ID in the schema/subschema.

Action: Change the XSETLINE parameters or supply a proper STATFILE.

W998 IMPLEMENTATION DELAYED FOR description

Reason: The feature described has not been implemented. The feature will be available in future versions of CA-IDMS/Schema Mapper.

Action: Do not try to use the feature.

E001 MISSING KEYWORD keyword

Reason: The indicated keyword is required.

Action: Add the keyword to the parameter statement.

E002 SCHEMA schema-name VERSION value COULD NOT BE FOUND

Reason: The designated schema name is misspelled or not in the dictionary. When ANY is given as the version number in this message, the designated schema could not be found with any version number in the dictionary.

Action: Check the spelling of the schema name and verify the version number specified.

E003 SUBSCHEMA subschema-name COULD NOT BE FOUND

Reason: The designated subschema name is misspelled or not defined for the requested schema and version.

Action: Check the spelling of the subschema name and verify it for the requested schema and version.

E004 PROCESS STATEMENT MISSING

Reason: The PROCESS statement is not included or is not the first statement.

Action: Verify that the PROCESS statement is in the input and that it is the first statement supplied.

E005 SECOND RECORD NAME MISSING IN LOCATE STATEMENT

Reason: The second record name in the LOCATE parameter statement is missing.

Action: Add the second record name to the LOCATE statement.

E006 INVALID SCHEMA VERSION NUMBER

Reason: The schema version number in the PROCESS parameter statement is not valid.

Action: Check the syntax of the schema version number on the PROCESS statement.

E007 CHARACTER character INVALID OR OUT OF CONTEXT

Reason: A character such as an exclamation point, which is not legal in any statement, or a character such as a hyphen, which is only legal within a name, is used. The character is skipped and syntax analysis continues. The erroneous statement appears immediately before this message.

Action: Check the statement for a character that is invalid or used out of context.

E008 OPTIONS STATEMENT OUT OF ORDER

Reason: The OPTIONS parameter statement is used but does not immediately follow the PROCESS statement.

Action: Place the OPTIONS statement immediately following the PROCESS statement.

E009 RECLINE STATEMENT OUT OF ORDER

Reason:

1. The RECLINE parameter statement is before the PROCESS, OPTIONS, or CHARDEF statement.
2. The RECLINE statement is after the SETLINE, INCLUDE, or LOCATE statements.

Action:

1. Verify that the RECLINE statement immediately follows the PROCESS, OPTIONS, and CHARDEF (if present) statements.

-
2. Verify that the RECLINE statement immediately precedes the SETLINE, INCLUDE, and LOCATE statements (if present).

E010 NAME REQUIRED INSTEAD OF symbol

Reason: A name is required in place of the indicated symbol. Either the name is missing or its syntax is incorrect. The erroneous statement appears immediately before this message.

Action: Verify that a name is in the indicated position and check its syntax.

E011 KEYWORD keyword REQUIRED INSTEAD OF symbol

Reason: A keyword is required at the location of the indicated symbol. The erroneous statement appears immediately before this message.

Action: Check the syntax of the statement for a missing keyword.

E012 UNKNOWN RECLINE FIELD NAME field-name

Reason: A field name specified in the RECLINE parameter statement is not predefined by CA-IDMS/Schema Mapper.

Action: Check the syntax of the field name and verify that it is a name known to CA-IDMS/Schema Mapper.

E013 UNKNOWN SETLINE FIELD NAME field-name

Reason: A field name specified in the SETLINE parameter statement is not predefined by CA-IDMS/Schema Mapper.

Action: Check the syntax of the field name and verify that it is a name known to CA-IDMS/Schema Mapper.

E014 LAYOUT SPACE EXHAUSTED

Reason: The data structure diagram exceeds the maximum layout space allowed (26 by 26 pages). The incomplete diagram is printed and the program terminates.

Action: Divide the schema into subschema sections and resubmit the job.

E015 UNKNOWN STATEMENT TYPE BEGINNING WITH symbol

Reason: A parameter statement is not recognizable.

Action: Check the first word of the statement.

E016

Reason: This message is reserved for technical support purposes.

Action: Contact Computer Associates Technical Support.

E017

Reason: This message is reserved for technical support purposes.

Action: Contact Computer Associates Technical Support.

E018

Reason: This message is reserved for technical support purposes.

Action: Contact Computer Associates Technical Support.

E019

Reason: This message is reserved for technical support purposes.

Action: Contact Computer Associates Technical Support.

**E020 PRODUCT INTEGRITY ERROR--NOTIFY PRODUCT
SUPPORT--error description**

Reason: An internal integrity check in the program failed. The message is followed by output (printed without page ejects) that helps Computer Associates Technical Support personnel locate the error.

Action: Notify Computer Associates Technical Support and forward a full copy of the Audit Report to them.

E021 FIRST RECORD NAME MISSING IN LOCATE STATEMENT

Reason: The first record name in the LOCATE parameter statement is missing. The erroneous statement appears immediately before this message.

Action: Add the first record name to the LOCATE statement.

E022 INTEGER REQUIRED INSTEAD OF symbol

Reason: An integer is required in place of the indicated symbol. The integer is missing or its syntax is incorrect. The erroneous statement appears immediately before this message.

Action: Verify that an integer is in the indicated position and check its syntax.

E023 COMMA REQUIRED INSTEAD OF symbol

Reason: A comma is required at the location of the indicated symbol. The erroneous statement appears immediately before this message.

Action: Check the syntax of the statement for a missing comma.

E024 name IS NOT A RECORD

Reason: The indicated name is not a record in the schema or subschema being processed. Instead, the name identifies a set or index. Only record names can be used in this position of the statement. The erroneous statement appears immediately before this message.

Action: Delete the statement or replace the name with a valid record name.

E025 UP/DOWN/LEFT/RIGHT REQUIRED INSTEAD OF symbol

Reason: Either UP, DOWN, LEFT, or RIGHT is required at the location of the designated symbol in a LOCATE statement.

Action: Verify that UP, DOWN, LEFT, or RIGHT is in the LOCATE statement and check the syntax.

E026 UNKNOWN NAME name-text

Reason: The indicated name is not in the schema or subschema being processed. The name appears in the statement immediately preceding this message.

Action: Delete the statement or use a valid name.

E027 INVALID INTEGER number

Reason: This error can be caused by a number that is too large.

Action: Check the range of legal values for the number in this parameter statement.

E028 BIND RUN-UNIT FAILED

Reason: CA-IDMS/Schema Mapper cannot perform the bind for this run-unit. This error is most likely caused by an unknown dictionary name. It can also occur if CA-IDMS is not available. The normal CA-IDMS error status is printed.

Action: Verify the dictionary name and check the syntax. Verify that CA-IDMS is available.

E029 UNKNOWN KEYWORD keyword

Reason: A keyword is unknown. As an example, TURKEY is unknown to CA-IDMS/Schema Mapper in the statement PROCESS=SCH,TURKEY=7. The first seven positions of the unknown keyword are shown.

Action: Verify that the keyword is part of CA-IDMS/Schema Mapper's syntax.

E030 UNKNOWN VALUE value

Reason: A value is unknown. As an example, NEVER is an unknown value in the statement OPTIONS, IXSET=NEVER.

Action: Verify that the value is part of CA-IDMS/Schema Mapper's syntax.

E031 IXCORNER VALUE MUST BE EITHER 2 OR 4

Reason: An index corner value other than 2 or 4 is in the OPTIONS parameter statement. The IXCORNER value must be either 2 or 4; no other values are allowed.

Action: Change the IXCORNER value in the OPTIONS statement to either 2 or 4.

E032 BORDER VALUE MUST BE BETWEEN 2 AND 50

Reason: A border space value less than 2 or greater than 50 is in the OPTIONS parameter statement. The BORDER value must be between 2 and 50, inclusive; no other values are allowed.

Action: Change the BORDER value in the OPTIONS statement to a number between 2 and 50, inclusive.

E033 DPAGELEN VALUE MUST BE BETWEEN 33 AND 132

Reason: A diagram page length value less than 33 or greater than 132 is in the OPTIONS parameter statement. The DPAGELEN value must be between 33 and 132, inclusive; no other values are allowed.

Action: Change the DPAGELEN value in the OPTIONS statement to a number between 33 and 132, inclusive.

E034 SCHEMA NAME NOT SPECIFIED

Reason: A schema name is not specified in the PROCESS parameter statement.

Action: Add the schema name to the PROCESS statement.

E035 SUBSCHEMA REQUESTED BUT SUBSCHEMA NAME NOT SPECIFIED

Reason: A subschema name is not specified in the PROCESS parameter statement with PROCESS=SUBSCHEMA specified.

Action: Add the subschema name to the PROCESS statement.

E036 INVALID PROCESS VALUE value

Reason: The PROCESS keyword value in the PROCESS parameter statement is invalid.

Action: Check the syntax of the PROCESS keyword value indicated.

E037

Reason: This message is reserved for technical support purposes.

Action: Contact Computer Associates Technical Support.

E038 PROCESS STATEMENT OUT OF ORDER

Reason: Optional parameter statements precede the PROCESS parameter statement.

Action: Place the PROCESS statement before optional parameter statements.

E039 LOCATE STATEMENT OUT OF ORDER

Reason: A LOCATE parameter statement is out of sequence.

Action: Check to see that all LOCATE statements are entered in a group after all of the other parameter statements.

E040 INVALID IMAGE VALUE value

Reason: The image value indicated is not valid.

Action: Check the validity of the IMAGE value.

E042 INVALID DPEJECT VALUE value

Reason: The diagram page eject value indicated is not valid.

Action: Check the validity of the DPEJECT value.

E043

Reason: This message is reserved for technical support purposes.

Action: Contact Computer Associates Technical Support.

E044

Reason: This message is reserved for technical support purposes.

Action: Contact Computer Associates Technical Support.

E045 NUMBER MISSING IN LOCATE STATEMENT

Reason: The keywords UP, DOWN, LEFT, and RIGHT must always be followed by a number in a LOCATE parameter statement.

Action: Add a number to the LOCATE statement.

E046 AT MOST ONE VERTICAL DISPLACEMENT MAY APPEAR IN A LOCATE STATEMENT

Reason: Multiple vertical displacements, as indicated with the UP and DOWN keywords, cannot be used in a LOCATE parameter statement.

Action: Delete the extra vertical displacements from the LOCATE statement.

E047 AT MOST ONE HORIZONTAL DISPLACEMENT MAY APPEAR IN A LOCATE STATEMENT

Reason: Multiple horizontal displacements, as indicated with the LEFT and RIGHT keywords, cannot be used in a LOCATE parameter statement.

Action: Delete the extra horizontal displacements from the LOCATE statement.

E048 UP/DOWN/LEFT/RIGHT MISSING BEFORE number

Reason: One of the four direction keywords must appear before each number in a LOCATE parameter statement.

Action: Add the appropriate direction keyword before the indicated number in the LOCATE statement.

E049 EXTRA IGNORED IN type-of STATEMENT (extra-characters)

Reason: Extra characters follow the required syntax of a parameter statement. The extra characters are ignored. The kind of statement is indicated by type-of, which may be LOCATE, INCLUDE, RECLINE, or SETLINE. The erroneous statement immediately precedes this message.

Action: Delete the extra characters in the indicated statement.

E050 FIRST LOCATE STATEMENT CANNOT HAVE DISPLACEMENT CLAUSES

Reason: The first LOCATE statement entered cannot include the direction keywords UP, DOWN, LEFT, or RIGHT.

Action: Revise the first LOCATE statement so it is in this form: LOCATE record-name

E051 FIRST LOCATE STATEMENT CANNOT HAVE 'FROM' CLAUSE

Reason: The first LOCATE statement in a series of LOCATEs cannot include a FROM clause.

Action: Revise the first LOCATE statement so it is in this form: LOCATE record-name

E052 DISPLACEMENT CLAUSE MISSING IN LOCATE STATEMENT

Reason: At least one displacement clause (UP/DOWN/LEFT/RIGHT number) must appear in all but the first LOCATE parameter statement you enter.

Action: Add a displacement clause to the LOCATE statement or delete the incomplete statement.

E053 DISPLACEMENT VALUE number OUT OF RANGE

Reason: A number following UP/DOWN/LEFT/RIGHT in a LOCATE parameter statement you entered is not between 1 and 10,000, inclusive.

Action: Change the displacement value in the LOCATE statement to a valid number.

E054 END-OF-FILE ENCOUNTERED WHEN CONTINUATION WAS EXPECTED

Reason: The syntax of a parameter statement indicated that the statement continued on another line; however, CA-IDMS/Schema Mapper encountered an end-of-file before finding the rest of the statement.

Action: Add the line(s) necessary to complete the statement or remove the comma that indicates continuation.

E055 THE REQUESTED SCHEMA OR SUBSCHEMA DOES NOT CONTAIN ANY RECORDS

Reason: The schema or subschema specified on the PROCESS parameter statement does not contain any records. A schema/subschema must contain at least one record before CA-IDMS/Schema Mapper can diagram it.

Action: Verify that you have correctly specified the schema/subschema. Specify a different schema/subschema in the PROCESS statement. If you specified the schema/subschema you wanted, but it does not contain any records, add one or more records to it.

E056 DPAGEWID VALUE MUST BE BETWEEN 33 AND 132

Reason: A diagram page width value less than 33 or greater than 132 is in the OPTIONS parameter statement. The DPAGEWID value must be between 33 and 132, inclusive; no other values are allowed.

Action: Change the DPAGEWID value in the OPTIONS statement to a number between 33 and 132, inclusive.

E057 INVALID COMPRESS VALUE value

Reason: The COMPRESS value specified in the OPTIONS parameter statement is invalid.

Action: Check the validity of the COMPRESS value indicated.

E058 INPUT FILE IS EMPTY. CHECK SPELLING OF NAME GIVEN FOR SYSIPT

Reason: The file or dataset used for the parameter statements input file either was completely empty or did not exist.

Action: Check the spelling of the file name or the dataset name used for SYSIPT. If the spelling of the name is correct, and the file or dataset exists, check the content of it.

E059 INCLUDE STATEMENT OUT OF ORDER

Reason: The INCLUDE statement must appear after any SETLINE statements and before any LOCATE statements.

Action: Check the order of parameter statements in the input file (SYSIPT).

E060 kind-of NAME MISSING IN INCLUDE STATEMENT

Reason: The INCLUDE statement preceding this message does not contain a name. The kind-of name is AREA, RECORD, SET, or INDEX.

Action: Check the syntax of the statement and add a name to it.

E061 name IS NOT AN AREA

Reason: The indicated name is not an area in the schema or subschema being processed. Instead, the name identifies a record, set, or index. You may use only area names in this position of the statement. The erroneous statement appears immediately before this message.

Action: Delete the statement or replace the name with a valid area name.

E062 COMMA MISSING BEFORE symbol

Reason: A comma is missing before the indicated symbol. The erroneous statement precedes this message. For RECLINE and SETLINE statements, the statement in error may be two lines back.

Action: Check the syntax of the statement for a missing comma.

E063 INTEGER MISSING FROM BLANK FIELD SPECIFICATION

Reason: A blank field was requested in a RECLINE or SETLINE statement, but no length was given for the field.

Action: Check the syntax of the statement and add the missing number.

E064 FIELD NAME MISSING IN RECLINE STATEMENT

Reason: Either a previous RECLINE statement contains two commas in a row, or a RECLINE statement ends with a comma.

Action: Check the syntax of the RECLINE statement and add the missing field name or delete the extra comma.

E065 FIELD NAME MISSING IN SETLINE STATEMENT

Reason: Either a previous SETLINE statement contains two commas in a row, or a SETLINE statement ends with a comma.

Action: Check the syntax of the SETLINE statement and add the missing field name or delete the extra comma.

E066 KEYWORD BLANK REQUIRED BEFORE number

Reason: The keyword BLANK must precede a number in a RECLINE or SETLINE field.

Action: Add the missing keyword or delete the number.

E067 SPACE MISSING IN field-name

Reason: A space must separate the keyword BLANK from the number that specifies the length of the field.

Action: Add the missing space or delete the field name.

E068 MORE THAN 20 BLANK FIELDS REQUESTED FOR kind-of-text

Reason: Only 20 blank fields may be requested in a series of RECLINE or SETLINE statements. Kind-of-text indicates whether the excess fields were requested for a record block or a set description.

Action: Delete the extra blank fields.

E069 WIDTH LIMIT EXCEEDED BY field-name FOR kind-of-text

Reason: The sum of the field sizes for each line of a record block or set description must be no more than 120. Be sure to allow for a space between adjacent fields. Kind-of-text indicates whether this was for a record block, or a set description.

Action: Check the RECLINE or SETLINE statement preceding this message. Either delete a field or move a field to another statement.

E070 SETLINE STATEMENT OUT OF ORDER

Reason:

1. The SETLINE parameter statement is before the PROCESS, OPTIONS, CHARDEF, or RECLINE statement.
2. The SETLINE statement is after the INCLUDE or LOCATE statements.

Action:

1. Verify that the SETLINE statement immediately follows the PROCESS, OPTIONS, CHARDEF, and RECLINE (if present) statements.
2. Verify that the SETLINE statement immediately precedes the INCLUDE and LOCATE statements (if present).

E071 SET FIELD field-name USED IN A RECLINE STATEMENT

Reason: The indicated field may only be used in SETLINE statements.

Action: Remove the field from the preceding RECLINE statement.

E072 RECORD FIELD field-name USED IN A SETLINE STATEMENT

Reason: The indicated field may only be used in SETLINE statements.

Action: Remove the field from the preceding RECLINE statement.

E073 TOO MANY RECLINE STATEMENTS

Reason: More than 14 RECLINE statements have been used.

Action: Remove the extra RECLINE statements.

E074 TOO MANY SETLINE STATEMENTS

Reason: More than 9 SETLINE statements have been used.

Action: Remove the extra SETLINE statements.

E075 parameter-name FIELD field-name ALREADY IN USE

Reason: Indicated field name appeared twice in a group of RECLINE or SETLINE statements.

Action: Remove extra occurrences of the indicated field name.

E076 kind-of PARENTHESIS MISSING IN SUBSCRIPT

Reason: The RECLINE or SETLINE statement preceding this message is missing a parenthesis in a subscript. The kind-of parenthesis is LEFT or RIGHT.

Action: Check the syntax of the statement and add the appropriate parenthesis.

E077 INVALID SUBSCRIPT VALUE value

Reason: The indicated subscript value is invalid. A subscript can be an integer from 1 to 256 or the letter N.

Action: Check the syntax of the indicated subscript and correct it.

E078 MISSING SUBSCRIPT VALUE

Reason: The RECLINE or SETLINE statement preceding this message contains parentheses but no subscript value.

Action: Check the syntax of the statement and add the subscript value or remove the parentheses.

E079 SUBSCRIPT NOT ALLOWED FOR field-name FIELD

Reason: Subscripts may only be used with the LOC-CTRL and SET-ORDER fields.

Action: Check the syntax of the RECLINE or SETLINE statement preceding this message and remove the erroneous subscript.

E080 FIELD NAME MISSING BEFORE LEFT PARENTHESIS

Reason: Subscripts, which begin with a left parenthesis, must follow the LOC-CTRL field in a RECLINE statement or the SET-ORDER field in a SETLINE statement.

Action: Check the syntax of the RECLINE or SETLINE statement and add the missing field name or remove the extra subscript value.

E081 RIGHT PARENTHESIS OUT OF PLACE

Reason: A right parenthesis may be used only as part of a subscript in a RECLINE or SETLINE statement. CA-IDMS/Schema Mapper may have encountered a subscript where only a simple number was expected (e.g., with the BLANK field).

Action: Check the syntax of the RECLINE or SETLINE statement and remove the right parenthesis or correct its position.

E082 LEFT PARENTHESIS OUT OF PLACE

Reason: A left parenthesis may be used only as part of a subscript in a RECLINE or SETLINE statement. CA-IDMS/Schema Mapper may have encountered a subscript where only a simple number was expected (e.g., with the BLANK field).

Action: Check the syntax of the RECLINE or SETLINE statement and remove the left parenthesis or correct its position.

E083 SUBSCRIPT VALUE number OUT OF RANGE FOR field-name FIELD

Reason: The indicated subscript value is either too large or too small for the indicated field. The valid range of subscript values is 1 through 256.

Action: Correct the invalid subscript value.

E084 CHARDEF STATEMENT OUT OF ORDER

Reason:

1. The CHARDEF parameter statement is before the PROCESS or OPTIONS statement.
2. The CHARDEF statement is after the DRECLINE, XRECLINE, DSETLINE, XSETLINE, INCLUDE, or LOCATE statements.

Action:

1. Verify that the CHARDEF statement immediately follows the PROCESS and OPTIONS (if present) statements.
2. Verify that the CHARDEF statement immediately precedes the DRECLINE, XRECLINE, DSETLINE, XSETLINE, INCLUDE, and LOCATE statements (if present).

E085 KEYWORD keyword MUST APPEAR ON CHARDEF STATEMENT

Reason: All character definition parameters must be given in the CHARDEF statement. These parameters are not allowed in the OPTIONS statement. The first seven positions of the keyword are shown.

Action: Move the indicated keyword and its associated value to the CHARDEF statement.

E086 THE schema-or-subschema ERROR FLAG IS ON

Reason: This message is turned on by the schema or subschema compiler when it detects an error during a compile. The dictionary may only contain part of the schema or subschema.

Action: Correct the error(s) and recompile the schema or subschema.

E087 OOAK-012 RECORD NOT FOUND

Reason: CA-IDMS/Schema Mapper did not find an OOAK-012 record in the requested dictionary.

Action: Check the integrity of the dictionary.

Glossary

border. The number of character spaces around the perimeter of each record block in the data structure diagram. A minimum border of two character spaces is needed for set connections and arrows; the maximum is 50 character spaces. A two-character border puts at least four character spaces between any two record blocks.

column header. A page with an easily-recognized format that you use as a reference to burst and align pages for wallpapering the CA-IDMS/Schema Mapper data structure diagram. Column headers separate each column of the diagram when it is printed out on successive pages.

coordinate position. A combination of two numbers, which appears in the Cross-Reference Report, that is used to find the exact location of a record block in a CA-IDMS/Schema Mapper diagram. The numbers refer to the position of the upper left corner of each record block in the diagram. The position is numbered in units of character spaces, with the upper left corner of the diagram being the origin (0,0). The first number tells how many character spaces the record block is from the left side of the diagram. The second number tells how many character spaces the record block is from the top of the diagram.

Cross-Reference Report. A report that contains the descriptions of sets, the name of each record, and location (coordinate position and page identifier) of each record block in the CA-IDMS/Schema Mapper data structure diagram. The report includes the name and unique set number of each set and index in the diagram. And it contains the names and locations, in the diagram, of the OWNER and MEMBER records of each set.

data structure diagram. A graphic representation of a CA-IDMS schema or subschema.

format. The textual content and textual organization of record block, set, and index information in the Cross-Reference Report or in the CA-IDMS/Schema Mapper data structure diagram. Also, the graphic components of the diagram, such as the characters used to draw record blocks, set connections, and arrows. You can control the format with the optional OPTIONS, CHARDEF, DRECLINE, XRECLINE, DSETLINE, and XSETLINE statements; or CA-IDMS/Schema Mapper's default format specifications can determine all or part of the format.

index. An index is represented by a uniquely numbered diagonal line, extending from a record block, in the

CA-IDMS/Schema Mapper data structure diagram. (See set number.) Indexes are also listed in the Cross-Reference Report.

layout. The arrangement of record blocks, sets, and indexes in a CA-IDMS/Schema Mapper data structure diagram. You can control the layout with the optional LOCATE parameter statement. CA-IDMS/Schema Mapper always does some amount of layout, by at least drawing set connections. CA-IDMS/Schema Mapper does complete automatic layout of all record blocks, sets, and indexes in the CA-IDMS/Schema Mapper data structure diagram, except when you manually position record blocks using LOCATE statements.

page identifier. A unique two-character (alphabetic) identifier that specifies the position of a (paper) page in a CA-IDMS/Schema Mapper data structure diagram. The first character identifies the page's column (which runs down the length of the diagram), and the second character identifies the page's row (which runs across the width of the diagram). The page identifier appears in the upper right corner of each page in the diagram and also next to record names in the Cross-Reference Report. The page identifier is useful as a reference when assembling the CA-IDMS/Schema Mapper data structure diagram and when looking for a record block in a diagram.

parameter statements. Statements used to specify parameters for an execution of CA-IDMS/Schema Mapper. The PROCESS parameter statement is mandatory. Any number of the optional statements--OPTIONS, CHARDEF, DRECLINE, XRECLINE, DSETLINE, XSETLINE, INCLUDE, and LOCATE--can also be used.

record block. A representation of an CA-IDMS record in a CA-IDMS/Schema Mapper data structure diagram. Record block descriptions can also be listed in the Cross-Reference Report. A record block contains various record fields. CA-IDMS/Schema Mapper supplies default format specifications. To tailor the data structure diagram you can use the optional DRECLINE statement. Use the optional XRECLINE statement to select and organize the record field information to be included in the Cross-Reference Report.

set. A group of record blocks that represent a CA-IDMS set and are connected to one another with set connection lines in the CA-IDMS/Schema Mapper data structure diagram. Each set is numbered in the diagram with a unique set number. Sets are also listed, with their

descriptions, in the Cross-Reference Report. Default format specifications are supplied for set appearances in the Cross-Reference Report and the data structure diagram. You can select and organize the fields to be included in the sets in the data structure diagram by using the optional DSETLINE statement, and in the Cross-Reference Report, by using the optional XSETLINE statement.

set connection. A physical line or series of lines, each with an arrow at the end that points toward a member of a set in the CA-IDMS/Schema Mapper data structure diagram. Set connections connect an owner of a set and its members.

set junction character. An uppercase letter O is used at a junction where a multi-member set connection line splits into more than one line.

set number. A unique number assigned by CA-IDMS/Schema Mapper to identify each set and index in a schema or subschema. Set numbers appear in the

CA-IDMS/Schema Mapper data structure diagram as part of set connection lines and index lines. For multi-member sets, the set number appears in the diagram next to the owner and each member. Set numbers are cross-referenced in the Cross-Reference Report.

tailored diagram. A data structure diagram as it appears after you have used at least one optional CA-IDMS/Schema Mapper parameter statement to modify the layout or format of the diagram that is generated by default (use of the PROCESS statement only).

Transfer File. An output file that reflects the layout and format of a CA-IDMS/Schema Mapper data structure diagram. The Transfer File contains parameter statements. You can use it as a time-saving device, as input, when recreating or modifying a CA-IDMS/Schema Mapper data structure diagram.

wallpapering. The process of bursting and assembling a CA-IDMS/Schema Mapper data structure diagram and (probably) hanging it on a wall for viewing.

Index

A

Audit report 1-20, 3-15
Automatic layout and format by default 1-10

C

CA-IDMS Dictionary 1-8
CA-IDMS/Schema Mapper's Four Outputs 1-10
CA-IDMS/Schema Mapper's Inputs 1-7
Changing page dimensions 2-13
CHARDEF statement 2-4, 2-15
CHARDEF syntax 2-15
Compressing unused space in the data structure diagram 2-14
Create Data Structure Diagrams Automatically 1-4
Creating blank fields and blank lines in record blocks 2-21
Creating blank fields and blank lines in record descriptions 2-29
Creating blank fields and blank lines in set descriptions 2-34, 2-40
Cross-reference report 1-19, 3-12
Cross-reference report field descriptions 3-12

D

Data structure diagram 1-10, 3-4
Data structure diagram, basic components of 3-4
Defining border space between record blocks 2-14
Diagram size 1-12
DRECLINE statement 2-5, 2-19
DRECLINE statement, how to use 2-21
DRECLINE syntax 2-20
DSETLINE statement 2-5, 2-33
DSETLINE statement, how to use 2-34
DSETLINE syntax 2-33

E

Example 1 4-4
Example 2 4-7
Example 3 4-9
Example 4 4-13
Example 5 4-15

G

Graphic considerations 2-21, 2-29, 2-34, 2-40, 2-49

I

INCLUDE AREA statement 2-5, 2-46
INCLUDE AREA statement, when to use 2-46
INCLUDE AREA syntax 2-46

L

Layout parameters, optional INCLUDE AREA statement 1-11
LOCATE statement 2-5, 2-47
LOCATE statement, when to use 2-48
LOCATE syntax 2-47

M

Multiple CALC keys, placement of 2-29
Multiple sort or index keys, placement of 2-35, 2-40

N

Notations Conventions and Syntax Rules 2-6

O

Optional LOCATE statement 1-11
Optional parameters, optional statements 1-11
OPTIONS statement 2-4, 2-12
OPTIONS syntax 2-12
Organization vii
OS/390 Environments 5-5
OS/390 JCL 5-5
OS/390 JCL, key to 5-5

P

Parameter statements 1-7
Placement of multiple CALC keys 2-22
Powerful Layout and Format Capabilities 1-6
PROCESS statement 2-4, 2-10
PROCESS syntax 2-10
Processing Environment 1-5
Purpose vii

S

Sample DRECLINE statements 2-23
Sample DSETLINE statements 2-36
Sample XRECLINE statements 2-30
Sample XSETLINE statements 2-42
Set linkage with areas not represented in diagram 2-14
Specifying print characters used to draw arrows 2-17
Specifying print characters used to draw index lines 2-18
Specifying print characters used to draw record blocks 2-16
Specifying print characters used to draw set connections 2-16
Specifying the positions of index 2-12
Storage requirements 5-4
System Requirements 5-4

XRECLINE statement, how to use 2-29
XRECLINE syntax 2-27
XSETLINE statement 2-5, 2-38
XSETLINE statement, how to use 2-40
XSETLINE syntax 2-38

T

Tailoring the cross-reference report 1-19
Tailoring the layout and format 1-10
Transfer file 1-14, 3-10
Transfer file statements 3-10

U

Using CA-IDMS/Schema Mapper's Parameters 2-4
Using the LOCATE statement most efficiently 2-48
Using the transfer file 1-14

V

VM/ESA Environments 5-12
VM/ESA EXEC 5-12
VM/ESA EXEC, key to 5-13
VSE/ESA Environments 5-8
VSE/ESA file assignments 5-8
VSE/ESA JCL 5-8
VSE/ESA JCL, key to 5-10

W

Wallpapering, assembling a data structure diagram 1-12
Wallpapering, used to assemble a data structure diagram 3-6
Ways to use the transfer file 1-16

X

XRECLINE statement 2-5, 2-27

